THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Scan matching for terrain mapping in open-pit mining

Felipe Antonio Donoso Aguirre

Master of engineering science
B. E. (Electrical engineering)

A thesis submitted for the degree of Master of Philosophy at
The University of Queensland in 2015
School of Mechanical & Mining Engineering

Abstract

This thesis is concerned with the problem of creating three-dimensional terrain maps using a high-density, high-rate LiDAR sensor mounted on a moving platform without the use of an external localization solution. In particular, the thesis is motivated by the problem of building terrain maps in open-pit mining environments from data produced by haul-truck mounted LiDAR sensors.

A foundation problem in map building from such data is the process of assembling together the point clouds generated by individual sensor scans into a common frame of reference. This falls into a broader class of well-studied problems called scan matching. Scan matching is most commonly solved by the Iterative Closet Point (ICP) method.

This thesis makes two contributions. The first contribution is a comparison of the different ICP variants for the candidate application. The literature is replete with different algorithms that can be used at different stages of the ICP process. A population of 20,736 ICP-variants drawn from methods proposed in the literature is compared. Whilst others have looked to compare different ICP-variants, this investigation is distinguished from these earlier works in three respects: i) its comprehensiveness; ii) the focus of its target application (the loosely structured terrain of open-cut mining); and iii) its emphasis on how accuracy, precision, and computational efficiency trade-off across different variants. The main finding of this investigation is that the geometry of the point cloud critically determines the quality of the scan match. Significantly, of the variants considered, none were found to simultaneous meet requirements on accuracy, precision, and efficiency, highlighting the need for better approaches.

The second contribution comes in the form of progress towards improving on the performance of established methods. For this, the thesis introduces the concept of "eigentropy" to quantify the geometric disorder or geometric information for points of a cloud. Eigentropy is conceptually similar to entropy as it appears in thermodynamics and information theory. Three novel methods are introduced to improve the performance of ICP-based scan-matching for loosely structured terrain. These methods are termed the: eigentropy filter, matching by normal deviation and unilateral eigentropy rejection, and they are based on the geometrical information content at a point. The results associated with these methods improve the overall performance of the ICP algorithm and five accurate, precise, efficient, and robust ICP variants are identified from an expanded population of 73,728. These five variants all use the methods introduced in this thesis.

Declaration by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the policy and procedures of The University of Queensland, the thesis be made available for research and study in accordance with the Copyright Act 1968 unless a period of embargo has been approved by the Dean of the Graduate School.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.

Publications during candidature

None.

Publications included in this thesis

None.

Contributions by others to the thesis

The data set presented in Chapter 3 were collected for a research project entitled 'Perception Sensor Capabilities', which was executed by the Smart Machines Group at the University of Queensland through the Cooperative Research Centre for Mining (CRCMining) with extensive support from Caterpillar Inc.

Statement of parts of the thesis submitted to qualify for the award of another degree

None.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor, Prof. Ross McAree, for the continuous support of my research, for his patience, motivation, wisdom and tireless assistance.

In the same way, I would like to thank to my co-advisor, Dr. Kevin Austin, for his technological assistance and thesis corrections. I am also indebted to my dear colleges: Tyson Phillips for his support in Matlab, thesis corrections and friendship; and Richard Hensel for reading my thesis.

This project would have been impossible without the support of CONICYT (Becas Chile) and CRCMining.

I also would like to thank my family for their endless caring and love. Particularly, to my parents for the infinite support of all my adventures.

Finally, to my caring, loving, and supportive Andrea: my deepest gratitude. Thanks for your selfless work on grammar corrections at the beginning of my research and your lovely companionship at the end to make my life easier and happier.

Keywords

mining automation, perception, terrain mapping, registration, ICP, eigentropy, velodyne

Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 091302, Automation and Control Engineering, 100%

Fields of Research (FoR) Classification

FoR code: 0913, Mechanical Engineering, 100%

Dedication

To my beloved family

Para mi amada familia

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis examines the problem of creating three-dimensional terrain maps using a high-density, high-rate LiDAR[1] sensor mounted on a moving platform without the use of an external localization solution. In particular, the thesis is concerned with the process of building terrain maps in open-pit mining environments with haul-truck mounted LiDAR sensors in support of automation of these machines.

1.1 Motivation

The mining industry is on a journey towards automation of mobile equipment and, among other things, this requires fast, accurate, robust and economic perception capabilities. The powerful LiDAR sensors and range cameras that have emerged in the last decade are driving much of the technology development in this field. An example of such sensors is the Velodyne® (Velodyne LiDAR Inc, 2008) depicted in Fig. 1.1.

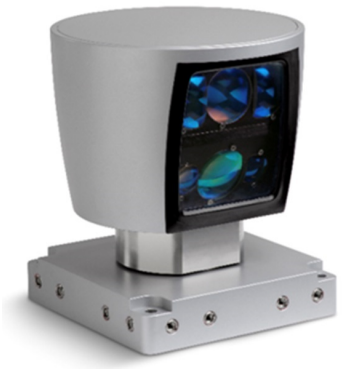


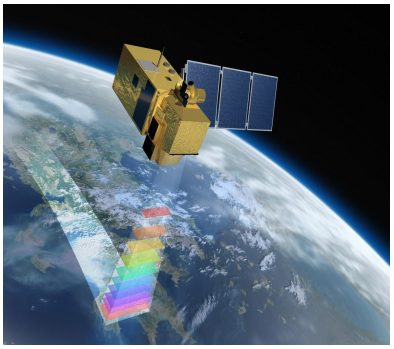
Figure 1.1: Velodyne HDL-64E LiDAR scanner.

Along with the growing complexity of autonomous mining machines, perception tasks have evolved

[1] LiDAR: Light Detection And Ranging.

to a rich set of activities in order to represent, interpret, and understand the environment through sensory information. However, building a representation of the environment by incorporating a sequence of acquired range sensor data into a common frame is probably the oldest and most fundamental perception task. This process is commonly known as *mapping* and the goal is to create a map of the environment that is able to be used by higher level automation functions to reason about the world and make control decisions.

There exist commercial solutions used in mining for digital terrain mapping targeted at mineral exploration, mine planning, and production reconciliation. Figure 1.2 shows the three most common commercial mapping solutions: satellite imaging, aerial photography, and terrestrial scanning. All are capable of producing accurate digital terrain maps and all are widely used in mining. However, these solutions, alone, are not suitable for automation applications which must take into account the constantly changing environment. The point cloud generated for a terrestrial scanner such as the Faro® unit depicted in Fig. 1.2c, for example. This system takes in the order of half an hour to construct a map. This is too slow to make the resulting terrain map useful for mobile equipment automation on its own, although it may well provide a useful base map that can be evolved and updated using information from real-time mapping technologies.



(a) Satellite images.

(b) Aerial photograph and laser scanning.

(c) Terrestrial scanner.

Figure 1.2: Three commercial mapping solutions.

Real-time mapping can be achieved by mounting range scanning sensors onto mobile equipment and building maps (or updating existing maps) from the resulting point cloud data. This general theme is the motivation for this thesis. The specific aim is to construct terrain maps from scan data produced by a Velodyne HDL-64E sensor mounted to a mining haul-truck that moves about the mine as part of its work.

Mapping by range scanning is intrinsically married with the localization of the sensor. If the position of the sensor is known, the data from consecutive scans can be coherently combined into a known spatial frame of reference in order to build a larger representation of the environment, commonly known as a *map*.

Mapping is most straightforward when the localization of the sensor is given, at any moment, by an external source. In cognate robotic applications, external localization can be, and often is, provided by the Global Navigation Satellite System (GNSS) (Johnson and Van Diggelen, 1998), complimented by information from other sensors such as inertial measurement units (IMU), odometers, and beacons (Fallon et al., 2012). A common configuration combines data from a GNSS receiver and a high-quality IMU using an extended Kalman filter to fuse satellite and inertial information into a six degree-of-freedom (6DOF) pose solution.

The Applanix POS LV 420 is an example of a commercially available system that combines GNSS solutions with data from a high quality IMU in what is called a tightly coupled architecture Applanix®. The resulting pose estimates are globally accurate to 0.02 m in ground plane coordinates, 0.05 m in vertical coordinates, 0.015 degrees in roll and pitch, and 0.02 degrees in yaw. This level of accuracy is sufficient to enable maps fit for automation purposes to be constructed.

A drawback of GNSS-based localization is the requirement for line-of-sight visibility to a sufficient number of satellites in order to maintain an accurate location fix (Kloos et al., 2004). Duff (2006) has noted, in the context of open-pit mining, that absolute localization techniques frequently struggle around the working areas of excavators, mainly because of the occlusion of satellite signals.

A high-end IMU with low drift and low intrinsic noise can in principle compensate for this by using an inertial-only pose solution during periods of GNSS outage and this is sometimes touted as a benefit of employing an external coupled GNSS/IMU solution for localization (Ford et al., 2001). However in practice, even with a very high quality inertial navigation solution, large errors manifest over short outage intervals, e.g. tens of seconds. These errors are due to: (i) the inherent integration involved in converting measurements of accelerations and angular rates into spatial pose estimates, a process that drifts in the presence of noise; and (ii) the broad spectrum vibrations that an inertial measurement unit experiences when fitted to a vehicle such as a haul-truck due, for example, to engine rumble and road unevenness which have magnitudes above the sensor noise levels and cause an inertial-only pose solution to diverge faster that they ordinarily would.

An alternative, complimentary method is to use information in overlapping scans from a range sensor as a basis for mapping and localization. *Prima facie* this seems to be a promising possibility, particularly with the recent availability of sensors such as the Velodyne HDL-64E sensor (Velodyne LiDAR Inc, 2008). This sensor has 64 emitter-detector pairs that provide in excess of one million range measurements per second over a 360 degree azimuth field of view at a resolution of 0.09 angular degrees and an elevation field of view of 26.8 degrees, with beam separation of 0.4 degrees. Range estimates are accurate to less than 0.05 m to a full range distance of 120 m. This sensor can complete 360 degree azimuth scans at up to 20 Hz. Consequently there is significant spatial overlap between scans at the speeds of operation of mining vehicles such as haul-truck (up to 60 kph), making these sensors good candidates for scan matching tasks.

1.2 Scan matching

The problem of localization and mapping, only with sensory information, can be addressed as two separate problems or as one combined problem. The idea of simultaneously seeking the position of the sensor and the construction of the map falls into an area known as Simultaneous Localization and Mapping (SLAM). Twenty years of developments in this field are summarized in Thrun et al. (2005).

The nexus point between localization and mapping is the geometrical relation of consecutive scans of the sensor. These geometrical relations are the foundation on which to join or match the scans in order to create the map, or obtain the odometry, by finding the transformation between scans. The scan matching is carried out by transforming the scan or point cloud into a new coordinate system of reference. Figure 1.3 shows an example of a point cloud rotation and translation needed to match the two clouds.

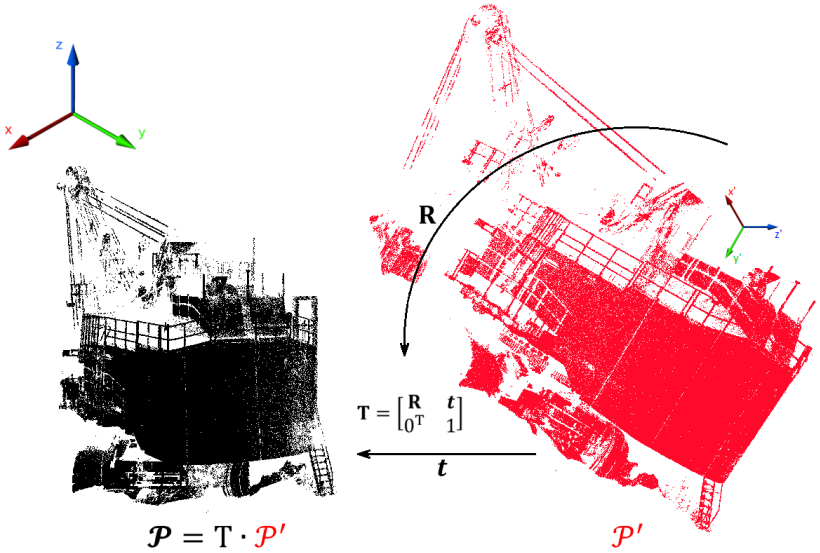


Figure 1.3: Transformation, T, of the cloud point scan, $\mathcal{P}'$, into a new coordinate system.

The role of scan matching in the broader process of mapping, is described well by Planitz et al. (2005), see Fig. 1.4. The four key steps are:

1. Correspondence: Finding correspondences of the points between two scans or between a scan and a reference map.

2. Minimization: Finding the rigid transformation T applied to the current scan $\mathcal{P}$ that minimizes the distance to the reference scan $\mathcal{Q}$.

3. Integration: Applying the transformation T to the current scan $\mathcal{P}$ to integrate with the reference map. Usually T is a composition of many transformations referenced to a common frame.

4. Actualization: Merging the transformed scan $\mathcal{P}(\mathcal{T})$ into the reference map.

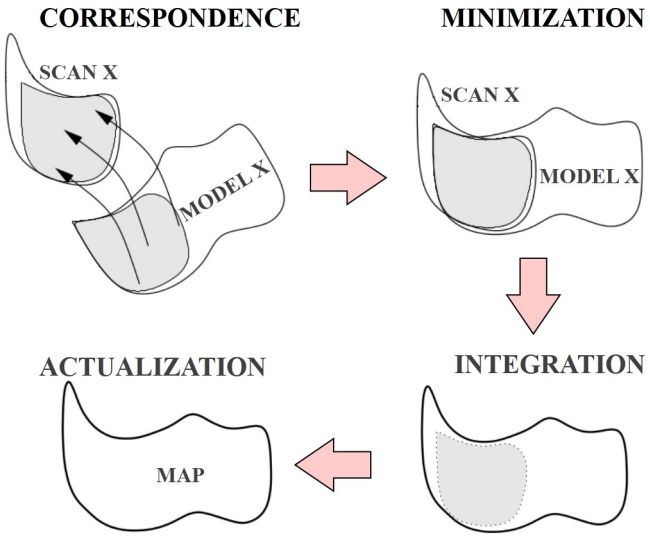


Figure 1.4: Simplified mapping process by Planitz et al. (2005).

The general problem of matching scans is well-studied, see for example Weiß and Puttkamer (1995); Lu and Milios (1997); Diosi and Kleeman (2005); Magnusson and Duckett (2005); Ye and Liu (2012); Biber and Straßer (2006). Approaches are usually based on the *Iterative Closest Point* (ICP) algorithm (Besl and McKay, 1992; Chen and Medioni, 1992) which is used to compute the transformation that brings two point clouds into "best" alignment by a two step process: (i) *correspondence*, the matching of overlapping data across the point clouds; and (ii) the *minimization* of a metric describing misalignment. In most applications these steps are iterated to improve alignment. Many variants of the ICP algorithm have been developed for 3D scan matching, focusing on improving the speed, accuracy, precision, and robustness of the method (Rusinkiewicz and Levoy, 2001; Salvi et al., 2007). The popularity of ICP algorithm derives from the simplicity of the method and the good performance that it has demonstrated across numerous applications.

Even though many ICP variants for scan matching have been developed and studied for over two decades, there is a gap in determining which variant should be used in a specific application. An ICP variant that performs well in one context may perform poorly in other contexts. The main focus of this thesis is on determining how best to scan match point clouds in mining environments, recognizing that there is no generally accepted ICP variant that is known to work well in this context.

1.3 Thesis objectives

The reader of this thesis is imagined to be the implementer of scan matching by ICP for mine terrain mapping, who is tasked with selecting an ICP variant from the many variants that stand as alternatives.

The specific aims of the thesis are to:

1. Evaluate existing scan matching methods based on ICP algorithms to identify those that perform best in an open-pit mining environment.
2. Identify the limitations of existing ICP variants including the challenges of scan matching in this context.
3. Propose variations or extensions that address these limitations and improve on performance to produce more effective scan matching.
4. Evaluate the performance of alternative methods and identify those that appear best suited to the application.

1.4 Thesis outline

The structure of the remainder of the thesis is as follows:

Chapter 2 gives an overview of the basic ICP method focussing on three variants known as point-to-point, point-to-plane, and generalized ICP. This chapter establishes necessary theoretical background for the thesis.

Chapter 3 describes the data set used for this study. Three different mining scenes are presented. The process to create the ground-truth used for the mapping is described. The methods used to compare different ICP variants are established.

Chapter 4 explores the performance of the many ICP variants that are potentially applicable to this work. Benefits and limitations of existing ICP variants are presented.

Chapter 5 presents three new ICP variants that aim to improve on existing methods. Five variants are identified whose average performance is accurate, robust, and efficient.

Chapter 6 summarizes the overall work of this thesis and recommends further work in the area of three-dimensional scan matching for terrain mapping.

Chapter 2

Scan matching by ICP

Scan matching is the process of aligning two range images or point clouds, using a rigid-body transformation, T, that minimizes the distance between them. This chapter reviews the basic iterative closest point (ICP) algorithm which is well established as the preferred scan matching method. In its bare-bones form, the algorithm consists of two steps that are applied iteratively. The first step establishes correspondences between the two point clouds. The second step computes the rigid-body transform that minimizes the error between correspondences. The focus of this chapter is on the second of these steps, i.e. minimization. The three most common methods used for minimization are described and the implementations of the solution of the minimization of distance function are presented. This chapter builds essential background material for the ideas covered in subsequent Chapters 4 and 5 where different variants of ICP are described and analyzed.

2.1 Iterative closest point

The starting point for exploring the iterative closest point algorithm is the definition of a point cloud, $\mathcal{P}$, as a collection of points, $\mathbf{p}_i$, in three dimensional Euclidean space,

$$\forall \mathbf{p_i} \in \mathcal{P}, \quad \mathbf{p}_i = \left\{ \; x_i, \quad y_i, \quad z_i \; \right\}.$$

Let the two point clouds to be matched be $\mathcal{P}^n$ and $\mathcal{Q}^m$ where n and m denote the cardinality of the clouds, that is, the number of points contained in each. Assume subsets of the points of each cloud can be paired and indexed (a process known as forming correspondences) so that $\hat{\mathbf{p}}_i$ corresponds to $\hat{\mathbf{q}}_i$. Denote the correspondence sets $\hat{\mathcal{P}}^N$ and $\hat{\mathcal{Q}}^N$ with N the number of points with established correspondences satisfying $N \leq n$. For sake of notation simplicity, $\mathbf{p}_i$ and $\mathbf{q}_i$ are considered to be a correspondence pair of $\hat{\mathcal{P}}^N$ and $\hat{\mathcal{Q}}^N$ hereafter.

Furthermore, assume the distance between the two point clouds $\mathcal{D}$ is defined by an error function,

$$E(T) = \mathcal{D}\left(\hat{\mathcal{Q}}^N, \hat{\mathcal{P}}^N\left(T\right)\right). \tag{2.1}$$

The rigid body transform T minimizing this error function satisfies,

$$T = \underset{T}{\operatorname{argmin}}\, \mathcal{D}\left(\hat{\mathcal{Q}}^N, \hat{\mathcal{P}}^N(T)\right). \tag{2.2}$$

Here T can be thought of as an optimization parameter applied to $\mathcal{P}^n$ to minimize the distance to $\mathcal{Q}^m$ given point correspondences $\hat{\mathcal{P}}^N$ and $\hat{\mathcal{Q}}^N$. T is composed of a 3×1 translation vector $\mathbf{t}$ in the direction of the Cartesian coordinate axes x, y and z, and a 3×3 rotation matrix R over the three axis. T has the following structure,

$$T = \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}.$$

In its most general form, the iterative closest point algorithm finds T by Algorithm 1.

Algorithm 1 Basic iterative closest point algorithm

Require: Two point clouds $\mathcal{P}^n$ and $\mathcal{Q}^m$. Seed transform T_0.

Ensure: A transformation T that aligns $\mathcal{P}^n$ to $\mathcal{Q}^m$.

1: $T \leftarrow T_0$
2: **while** not converged **do**
3: $\left(\hat{\mathcal{P}}^N, \hat{\mathcal{Q}}^N\right) \leftarrow \texttt{EstablishCorrespondances}(\mathcal{Q}^m, \mathcal{P}^n(T))$
4: $T \leftarrow \underset{T}{\operatorname{argmin}}\, \mathcal{D}\left(\hat{\mathcal{Q}}^N, \hat{\mathcal{P}}^N(T)\right)$
5: **end while**

Many variations to this algorithm have been proposed over the last twenty years. The variations of interest to us in this chapter are associated with different ways of completing Line 4, the so-called minimization step. Three approaches are in common use; point-to-point ICP (Besl and McKay, 1992), point-to-plane ICP (Chen and Medioni, 1992) and generalized ICP (Segal et al., 2009). Each is now reviewed in turn.

2.2 Point-to-point ICP

Let $\mathbf{d}_i$ be the Cartesian distance from a rotated and translated point of the input point cloud, $\tilde{\mathbf{p}}_i$, to the corresponding point $\mathbf{q}_i$,

$$\mathbf{d}_i = \mathbf{q}_i - \underbrace{(R\mathbf{p}_i + \mathbf{t})}_{\tilde{\mathbf{p}}_i}. \tag{2.3}$$

The distance between point clouds can be expressed as the average of the distance between points of the cloud by,

$$\mathcal{D} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{d}_i\|^2. \tag{2.4}$$

Substituting Eqns. 2.3 and 2.4 into Eqn. 2.1, allows the error function to be expressed in term of the distance between points as,

$$E\left(R,\mathbf{t}\right)=\frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{q}_i-\left(R\mathbf{p}_i+\mathbf{t}\right)\right\|^2. \tag{2.5}$$

Different strategies can be used to minimize Eqn. 2.5 in order to find R and $\mathbf{t}$. These are broadly classified as indirect and direct (Nüchter, 2009). Indirect methods are iterative and usually based on gradient descent. Direct methods are closed-form and generally more computationally efficient.

The generally preferred approach and the one used in later chapters of this thesis is based on singular value decomposition (SVD) and was developed by Arun et al. (1987). The method decouples calculation of the rotation and translation components. The principal assumption is that the position of points of $\hat{\mathcal{P}}^N$ and $\hat{\mathcal{Q}}^N$, at the optimal transformation, are coincident. This assumption is fundamental to explain the performance of ICP variants based in the point-to-point minimization. This assumption is given as,

$$\mathbf{q}_i=R\mathbf{p}_i+\mathbf{t},\qquad \forall i\in 1\ldots N. \tag{2.6}$$

The solution of the point-to-point minimization is derived using Eqn. 2.6 by obtaining the mean of $\mathbf{q}_i$,

$$\frac{1}{N}\sum_{i=1}^{N}\mathbf{q}_i=R\frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i+\cancel{\frac{1}{N}\sum_{i=1}^{N}\mathbf{t}}^{\mathbf{t}}. \tag{2.7}$$

Let the centroids of $\hat{\mathcal{P}}^N$ and $\hat{\mathcal{Q}}^N$ be $\mathbf{c}_p$ and $\mathbf{c}_q$ where,

$$\mathbf{c}_p=\frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i,\qquad \mathbf{c}_q=\frac{1}{N}\sum_{i=1}^{N}\mathbf{q}_i. \tag{2.8}$$

Rewriting Eqn. 2.7 with respect to $\mathbf{c}_p$ and $\mathbf{c}_q$,

$$\mathbf{c}_q=R\mathbf{c}_p+\mathbf{t}. \tag{2.9}$$

Now, referring $\mathbf{q}_i$ and $\mathbf{p}_i$ to their centroids,

$$\left\{\mathbf{p}'_i=\mathbf{p}_i-\mathbf{c}_p,\qquad \mathbf{q}'_i=\mathbf{q}_i-\mathbf{c}_q\right\},\qquad \forall i\in 1\ldots N.$$

where $\mathbf{p}'_i$ and $\mathbf{q}'_i$ define the new pair i centred in $\mathbf{c}_p$ and $\mathbf{c}_q$, respectively. Reordering Eqn. 2.2:

$$\left\{\mathbf{p}_i=\mathbf{p}'_i+\mathbf{c}_p,\qquad \mathbf{q}_i=\mathbf{q}'_i+\mathbf{c}_q\right\},\qquad \forall i\in 1\ldots N. \tag{2.10}$$

Substituting Eqn. 2.10 into Eqn. 2.5,

$$E\left(R,\mathbf{t}\right)=\frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{q}'_i-R\mathbf{p}'_i-\underbrace{\left(\mathbf{t}-\mathbf{c}_q+R\mathbf{c}_p\right)}_{\tilde{\mathbf{t}}}\right\|^2. \tag{2.11}$$

Equation 2.9 shows that the term $\tilde{\mathbf{t}}$ is zero at the optimal transformation. Now the quadratic error can be expressed as,

$$E\left(R,\mathbf{t}\right)=\frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{q}'_i-R\mathbf{p}'_i\right\|^2. \tag{2.12}$$

Therefore, the rotation and translation are decoupled and can be derived in two steps:

1. The rotation, R, can be obtained by minimizing the mean square error presented in Eqn. 2.12.
2. The translation, $\mathbf{t}$, can be calculated using the relation derived from Eqn. 2.9,

$$\mathbf{t}=\mathbf{c}_q-R\mathbf{c}_p.$$

Now using the method presented in Arun et al. (1987) to find the rotation,

$$R=VU^T \tag{2.13}$$

where matrices V and U are computed by singular value decomposition,

$$C=U\Lambda V^T \tag{2.14}$$

where C is the correlation matrix between $\mathbf{q}'_i$ and $\mathbf{p}'_i$. This 3×3 matrix is given by,

$$C=\sum_{i=1}^{N}\mathbf{p}'_i\mathbf{q}'^T_i=\begin{pmatrix}S_{xx}&S_{xy}&S_{xz}\\S_{yx}&S_{yy}&S_{yz}\\S_{zx}&S_{zy}&S_{zz}\end{pmatrix} \tag{2.15}$$

with:

$$S_{xx}=\sum_{i=1}^{N}q'_{x,i}p'_{x,i},\qquad S_{xy}=\sum_{i=1}^{N}q'_{x,i}p'_{y,i},\qquad S_{xz}=\sum_{i=1}^{N}q'_{x,i}p'_{z,i},\ldots.$$

2.3 Point-to-plane ICP

The point-to-plane distance corresponds to the closest distance from a point $\tilde{\mathbf{p}}_i$ to the best estimated plane within a given neighbourhood of $\mathbf{q}_i$. Figure 2.1 illustrates the geometrical relations required to calculate the minimum distance, $\mathbf{d}_i$, to a plane defined by the normal vector $\mathbf{n}_i$.

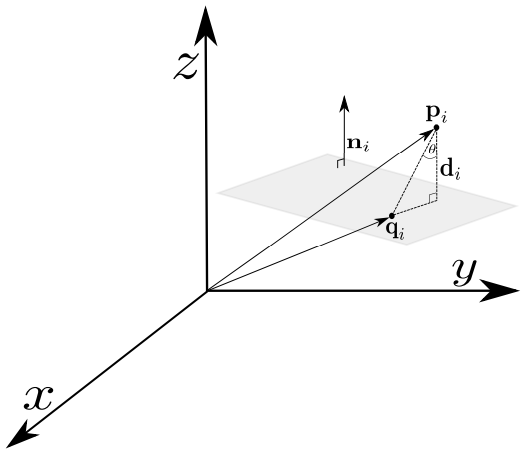


Figure 2.1: Distance from a point to a plane.

The norm of the minimum distance from a point, $\tilde{\mathbf{p}}_i$, to a plane is expressed by the dot product of the unitary normal vector of the plane, $\hat{\mathbf{n}}_i$, and the difference between $\mathbf{q}_i$ and $\tilde{\mathbf{p}}_i$. This relation is given by,

$$\|\mathbf{d}_i\| = \big(\mathbf{q}_i - \underbrace{(R\mathbf{p}_i + \mathbf{t})}_{\tilde{\mathbf{p}}_i}\big) \cdot \hat{\mathbf{n}}_i. \tag{2.16}$$

Substituting Eqn. 2.16 and Eqn. 2.4 into Eqn. 2.1, the error function for minimization of the point-to-plane distance can be expressed as,

$$E\left(R, \mathbf{t}\right) = \frac{1}{N} \sum_{i=1}^{N} \left\| \big(\mathbf{q}_i - (R\mathbf{p}_i + \mathbf{t})\big) \cdot \hat{\mathbf{n}}_i \right\|^2, \tag{2.17}$$

The aim of the algorithm is to find the R and $\mathbf{t}$ that, when are applied to $\mathbf{p}_i$, minimize the distance to the best fitted plane obtained in the neighbourhood of $\mathbf{q}_i$.

The rotational matrix, R, is a non-linear function of the rotational angles α, β and γ of the three axes x, y and z, respectively. It is assumed that rotational angles will be small, thus the $\cos(\theta)$ can be approximated to 1 and the $\sin(\theta)$ to θ. This is a reasonable assumption for angles smaller than 14° where the error of the approximation is less than 1%.

$$
\begin{aligned}
R_{x,\alpha} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \approx \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\alpha \\ 0 & \alpha & 1 \end{pmatrix} \\
R_{y,\beta} &= \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix} \approx \begin{pmatrix} 1 & 0 & -\beta \\ 0 & 1 & 0 \\ \beta & 0 & 1 \end{pmatrix} \\
R_{z,\gamma} &= \begin{pmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \approx \begin{pmatrix} 1 & \gamma & 0 \\ -\gamma & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.
\end{aligned} \tag{2.18}
$$

The rotational matrix can be obtained as,

$$R = R_{z,\gamma} R_{y,\beta} R_{x,\alpha}. \tag{2.19}$$

Thus, the rotational matrix can be approximated as,

$$R \approx \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix}. \tag{2.20}$$

Substituting Eqn. 2.20 into the point-to-plane error of Eqn. 2.17 and regrouping,

$$
\begin{aligned}
E(R, \mathbf{t}) = \frac{1}{N} \sum_{i=1}^{N} \Big\| &(\mathbf{q}_i - \mathbf{p}_i) \cdot \hat{\mathbf{n}}_i + \mathbf{t} \cdot \hat{\mathbf{n}}_i + \alpha \left(p_{i,y} n_{i,z} - p_{i,z} n_{i,y}\right) \\
&+ \beta \left(p_{i,z} n_{i,x} - p_{i,x} n_{i,z}\right) + \gamma \left(p_{i,x} n_{i,y} - p_{i,y} n_{i,x}\right) \Big\|^2.
\end{aligned} \tag{2.21}
$$

Defining the cross product between $\mathbf{p}_i$ and $\hat{\mathbf{n}}_i$ as,

$$\mathbf{v}_i = \mathbf{p}_i \times \hat{\mathbf{n}}_i$$

and grouping the rotation angles,

$$\mathbf{r} = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}.$$

Using $\mathbf{v}_i$ and $\mathbf{r}$ into the Eqn. 2.21, the error function can be written as,

$$E(R, \mathbf{t}) = \frac{1}{N} \sum_{i=1}^{N} \Big\| (\mathbf{q}_i - \mathbf{p}_i) \cdot \hat{\mathbf{n}}_i + \mathbf{t} \cdot \hat{\mathbf{n}}_i + \mathbf{r} \cdot \mathbf{v}_i \Big\|^2. \tag{2.22}$$

The values of $\mathbf{r}$ and $\mathbf{t}$ that minimize Eqn. 2.22 are obtained by setting the partial derivatives to zero,

$$\frac{\partial E}{\partial \mathbf{r}} = 0 \tag{2.23}$$

$$\frac{\partial E}{\partial \mathbf{t}} = 0. \tag{2.24}$$

Expanding and grouping the six equations obtained from 2.23 and 2.24 over $\alpha, \beta, \gamma, t_x, t_y$ and t_z:

$$\sum_{i=1}^{N} \begin{pmatrix} v_{i,x}v_{i,x} & v_{i,x}v_{i,y} & v_{i,x}v_{i,z} & v_{i,x}n_{i,x} & v_{i,x}n_{i,y} & v_{i,x}n_{i,z} \\ v_{i,y}v_{i,x} & v_{i,y}v_{i,y} & v_{i,y}v_{i,z} & v_{i,y}n_{i,x} & v_{i,y}n_{i,y} & v_{i,y}n_{i,z} \\ v_{i,z}v_{i,x} & v_{i,z}v_{i,y} & v_{i,z}v_{i,z} & v_{i,z}n_{i,x} & v_{i,z}n_{i,y} & v_{i,z}n_{i,z} \\ n_{i,x}v_{i,x} & n_{i,x}v_{i,y} & n_{i,x}v_{i,z} & n_{i,x}n_{i,x} & n_{i,x}n_{i,y} & n_{i,x}n_{i,z} \\ n_{i,y}v_{i,x} & n_{i,y}v_{i,y} & n_{i,y}v_{i,z} & n_{i,y}n_{i,x} & n_{i,y}n_{i,y} & n_{i,y}n_{i,z} \\ n_{i,z}v_{i,x} & n_{i,z}v_{i,y} & n_{i,z}v_{i,z} & n_{i,z}n_{i,x} & n_{i,z}n_{i,y} & n_{i,z}n_{i,z} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ t_x \\ t_y \\ t_z \end{pmatrix} = -\sum_{i=1}^{N} \begin{pmatrix} c_{i,x}(\mathbf{q}_i - \mathbf{p}_i) \cdot \hat{\mathbf{n}}_i \\ c_{i,y}(\mathbf{q}_i - \mathbf{p}_i) \cdot \hat{\mathbf{n}}_i \\ c_{i,z}(\mathbf{q}_i - \mathbf{p}_i) \cdot \hat{\mathbf{n}}_i \\ n_{i,x}(\mathbf{q}_i - \mathbf{p}_i) \cdot \hat{\mathbf{n}}_i \\ n_{i,y}(\mathbf{q}_i - \mathbf{p}_i) \cdot \hat{\mathbf{n}}_i \\ n_{i,z}(\mathbf{q}_i - \mathbf{p}_i) \cdot \hat{\mathbf{n}}_i \end{pmatrix}.$$

This is a linear equation of the form of $A\mathbf{x} = \mathbf{b}$. It can be solved using the Cholesky decomposition, $A = \mathrm{LL}^T$, where L is a lower triangular matrix.

The translation vector can be obtained straightforwardly from t_x, t_y and t_z. To obtain the rotation matrix, R, it is necessary to substitute α, β, γ into Eqn. 2.18 (using the exact rotational matrices) and calculate R by substituting $R_{z,\gamma}$, $R_{y,\beta}$ and $R_{x,\alpha}$ into Eqn. 2.19.

2.4 Generalized ICP

Segal et al. (2009) presents the Generalized-ICP as a generalization of the *total least squares* algorithm formulated by Estépar et al. (2004). ICP point-to-point and point-to-plane assume that points of both input and reference clouds have isotropic and identical probability distributions. Generalized-ICP minimization assumes that points of both point clouds are locally Gaussian distributions.

Points of the reference and input clouds are respectively described by the distributions $q_i \sim \mathcal{N}(\hat{q}_i, C_i^q)$ and $p_i \sim \mathcal{N}(\hat{p}_i, C_i^p)$, where C_i^q and C_i^p are the associated covariance matrix of each paired point. $\hat{q}_i$ and $\hat{p}_i$ are the associated position average of the points.

The Generalized-ICP method presented by Segal et al. (2009) uses the same assumption of equality of points in the optimal transformation assumed in the point-to-point distance. This assumption is depicted in Eqn. 2.6. In the same way as ICP point-to-point, the distance function is defined as,

$$d_i(T) = \mathbf{q}_i - R\mathbf{p}_i - \mathbf{t}, \tag{2.25}$$

where $\mathbf{q}_i$ and $\mathbf{p}_i$ are drawn from independent Gaussians distributions. Thus the probability distribution of $d_i(T)$ is:

$$d_i(T) \sim \mathcal{N}\left(\mathbf{c}_q - R\mathbf{c}_p - \mathbf{t}, C_i^q + RC_i^pR^T\right).$$

Because of the equality assumption of Eqn. 2.6, the probability distribution of the distance is shifted to the origin giving,

$$d_i(T) \sim \mathcal{N}\left(0, C_i^q + RC_i^pR^T\right) \tag{2.26}$$

Using *Maximum-Likelihood Estimation* (MLE) the transformation, T (i.e. R and $\mathbf{t}$), that maximizes the joint probability of the distance function, $d_i(T)$,

$$\begin{aligned} T &= \underset{T}{\operatorname{argmax}} \prod_{i=1}^{N} p(d_i(T)) \\ &= \underset{T}{\operatorname{argmax}} \sum_{i=1}^{N} \log\left(p(d_i(T))\right) \end{aligned} \tag{2.27}$$

The probability of the random variable d_i is given by the multivariate Gaussian distribution,

$$p\left(d_i(T)\right) = \frac{1}{\sqrt{(2\pi)^k \mid C_i^d \mid}} \exp\left(-d_i(T)^T (C_i^d)^{-1} d_i(T)\right) \tag{2.28}$$

where C_i^d is the covariance matrix associated with the distance $d_i(T)$,

$$C_i^d = C_i^q + RC_i^pR^T.$$

Substituting Eqn. 2.28 into Eqn. 2.27,

$$T = \underset{T}{\operatorname{argmax}} \sum_{i=1}^{N} -\log \mid C_i^d \mid - d_i(T)^T (C_i^d)^{-1} d_i(T) - k \log \pi, \tag{2.29}$$

where $-\log \mid C_i^d \mid$ and $-k \log \pi$ are constants. Maximizing a negative variable is the same as minimizing the same variable with the sign changed. Hence, Eqn. 2.29 can be expressed as,

$$T = \underset{T}{\operatorname{argmin}} \sum_{i=1}^{N} d_i(T)^T \left(C_i^q + RC_i^pR^T\right)^{-1} d_i(T). \tag{2.30}$$

The distance between point clouds can be generically expressed as,

$$\mathcal{D} = \frac{1}{N} \sum_{i=1}^{N} d_i(T)^T \left(C_i^q + RC_i^pR^T\right)^{-1} d_i(T). \tag{2.31}$$

Equation 2.31 provides a general form of the distance function that can be used for deriving the point-to-point and point-to-plane distances. Assuming $C_i^{\mathcal{Q}} = I$ and $C_i^{\mathcal{P}} = 0$ gives the point-to-point distance,

$$\begin{aligned}\mathcal{D} &= \frac{1}{N}\sum_{i=1}^{N} \mathbf{d_i}^T \mathbf{d_i} \\ &= \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{d}_i\|^2. \end{aligned} \tag{2.32}$$

Selecting $C_i^q = P_i^{-1}$ and $C_i^p = 0$, where P_i is the orthogonal projection of $\mathbf{p}_i$ into the best fitted plane within the neighbourhood of $\mathbf{q}_i$. Utilizing orthogonal projection properties ($P_i = P_i^2 = P_i^T$) allows Eqn. 2.31 to express the point-to-plane distance as,

$$\begin{aligned}\mathcal{D} &= \frac{1}{N}\sum_{i=1}^{N} \mathbf{d}_i{}^T P_i \mathbf{d}_i \\ &= \frac{1}{N}\sum_{i=1}^{N} \left(P_i\mathbf{d}_i\right)^T \left(P_i\mathbf{d}_i\right) \\ &= \frac{1}{N}\sum_{i=1}^{N} \|P_i\mathbf{d}_i\|^2 \end{aligned} \tag{2.33}$$

The projection P_i is not invertible because it is rank deficient. However, if P_i is approximated to an invertible matrix, the point-to-plane distance is approximately obtained.

Segal et al. (2009) proposes a constraint of the covariance matrices in order to increase the symmetry of the distance function. This constraint affects the covariance matrices by expressing the variance in the normal direction. Equation 2.34 shows the constrained covariance matrices $\tilde{C}_i^q$ and $\tilde{C}_i^p$ with small constant factor, e.

$$\begin{aligned}\tilde{C}_i^q &= C_i^q \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e \end{pmatrix} C_i^{qT} \\ \tilde{C}_i^p &= C_i^p \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e \end{pmatrix} C_i^{pT} \end{aligned} \tag{2.34}$$

The minimization of the distance function is achieved by iterative gradient descent methods where the minimization variables are the rotations α, β and γ and translations t_x, t_y and t_z.

2.5 Conclusions

This chapter presented the three most common methods used for ICP minimization. The solutions for minimizing the distance function where established for each method. These solutions are used in the evaluation of ICP variants of Chapter 4 and Chapter 5. The Chapter 3 presents the data scenes from a mining environment that are used for the evaluation of ICP variants considered in this thesis.

Chapter 3

Evaluation data and methodology

Several previous studies including Rusinkiewicz and Levoy (2001), Salvi et al. (2007), and Pomerleau et al. (2013) have noted that the performance of ICP methods depends critically on the characteristics of the point clouds being matched. For example, these characteristics include: perfect orthogonal planes in human made environments; irregular surfaces from rocky walls; regular grounds from machinery routes and; scattered points from vegetation. Indeed Rusinkiewicz and Levoy (2001) have suggested robustness might be best achieved by switching algorithms based on cloud characteristics. An algorithm that generally performs well still has appeal and subsequent chapters examine the performance of different variants.

This chapter describes the data set to be used in this evaluation and describes the methodology that will be used to compare the performance of different ICP variants.

3.1 Measurement data

The dataset used for this study is based on measurements from a Velodyne HDL-64E (Velodyne LiDAR Inc, 2008) sensor mounted to a Caterpillar 777B haul truck. The LiDAR has a 360 ° horizontal field of view and a 26.8 ° vertical field of view. The sensor was configured to scan at 20 Hz with an azimuth resolution of 0.09 °. Range measurements provided by the LiDAR unit have a standard deviation of 0.02 m and a maximum range of 120 m.

A ground-truth scan was established using the pose of the truck as measured by an Applanix POS LV 420 positioning system (Applanix Corp., 2008) that fuses an RTK GNSS solution with high accuracy inertial measurements. The Applanix navigation system is precise to 0.02 m in ground plane coordinates, 0.05 m in vertical coordinates, 0.015 ° in roll and pitch, and 0.02 ° in yaw.

The Velodyne LiDAR is registered to the navigation solution using the method described in Phillips et al. (2014) and has been assessed as being precise to 0.01 m and 0.05 ° for translation and rotation parameters respectively. The precision of registration is approximately an order of magnitude larger

than the precision of the navigation system.

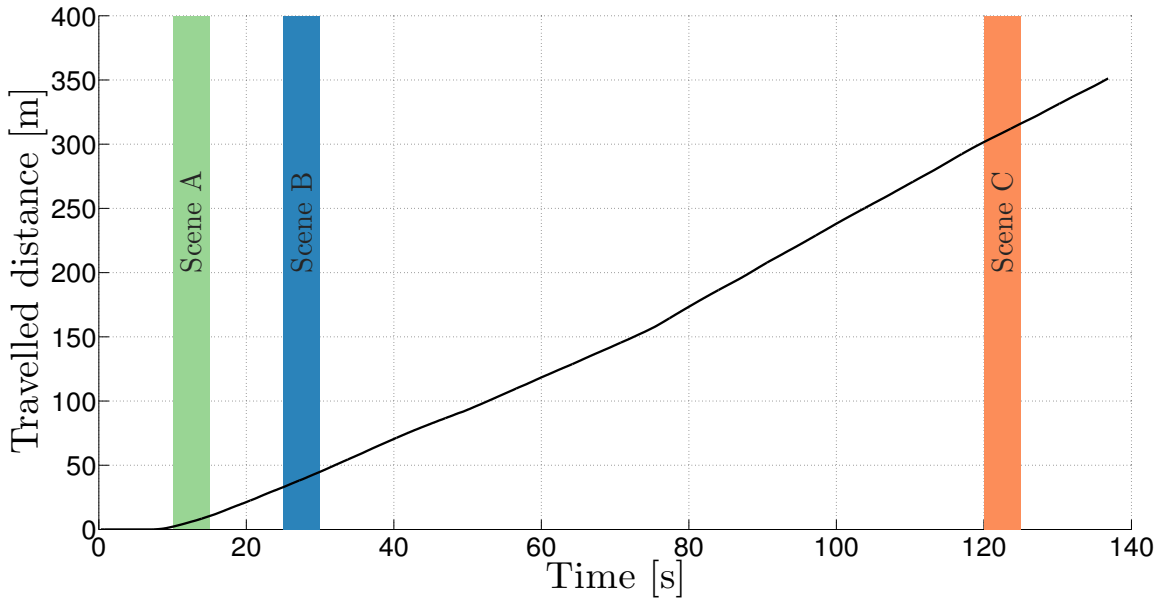
The LiDAR, GNSS/IMU navigation system and associated components were mounted to the front of a Caterpillar 777B haul truck, Fig. 3.1. The mounting configuration provided the Velodyne with a 180 ° field of view in front at the truck.

Figure 3.1: Mounting of Velodyne, and pose solution hardware on a Caterpillar 777B haul truck.

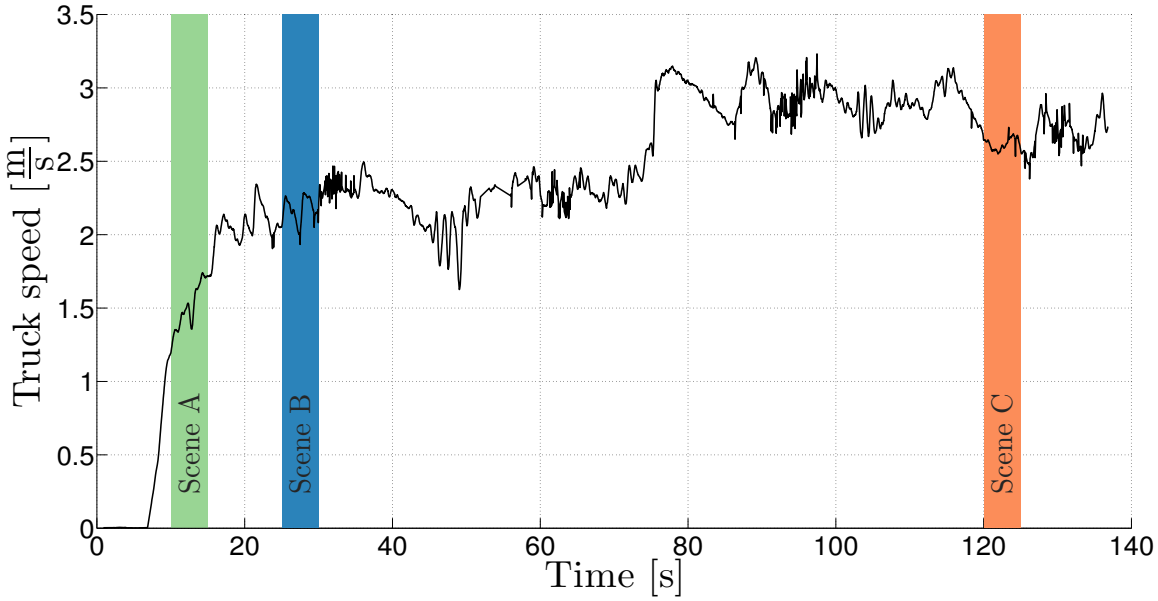
The Applanix registered Velodyne scans have been verified by comparison with a FARO Focus3D Terrestrial Laser Scanner (FARO Technologies Inc, 2010). The FARO sensor provides a one sigma range error of 0.3 mm at ranges of 10 m for 90% reflections with a beam divergence of 0.009 °. A dual axis inclinometer levels each scan with an accuracy of 0.015 °. A high density point cloud is generated with a possible vertical and horizontal step size of 0.009 ° and 0.036 ° respectively. Applanix registered Velodyne Scans were found to be within 0.2 m RMS of the corresponding FARO scan, giving confidence in the ground truth.

Data for this study has been collected during a period when the truck is moving along a section of haul-road that is approximately 350 meters long. Three sections of the haul-road are identified as scenes for analysis. The route starts at the working face, moves up a ramp and ends in a dumping

area. The total duration of the haul is 137 seconds. The three different scenes analysed have been chosen to be different. Each scene comprised five seconds of data and 100 scans from the sensor. Each scan contains approximately 84,500 points. While this data set is restricted it has chosen to be representative, and is considered sufficient to undertaking the analysis that follows. Figure 3.2 shows that the travelled distance and speed of the truck vary throughout the route.



(a) Travelled distance of the truck over the haul-route.



(b) Truck speed over the haul-route.

Figure 3.2: Truck travelled distance and speed across the three selected scenes.

Table 3.1 shows the travelled distance, truck speed and offset distance between scans for each scene.

Table 3.1: Scenes dynamically parameters.

Scene	Travelled distance [m]	Average speed [m/s]	Average distance between scans [m]
A	8.25	1.65	0.082
B	11.96	2.39	0.119
C	14.40	2.88	0.144

3.1.1 Data scenes

Figure 3.3 shows a top view of the truck operating environment with three selected scenes circled. These are considered representative of a haul-truck route, from the working face to a stockpile area. Details of the scenes are depicted below by a representative photograph of the work area and the consolidation of 100 scans (5 seconds of data at a 20 Hz scan rate) registered against the pose solution. These 100 scans mapped to the world frame via the pose solution are taken to constitute the ground truth. The spatial extent of each scene is up to 100 m.

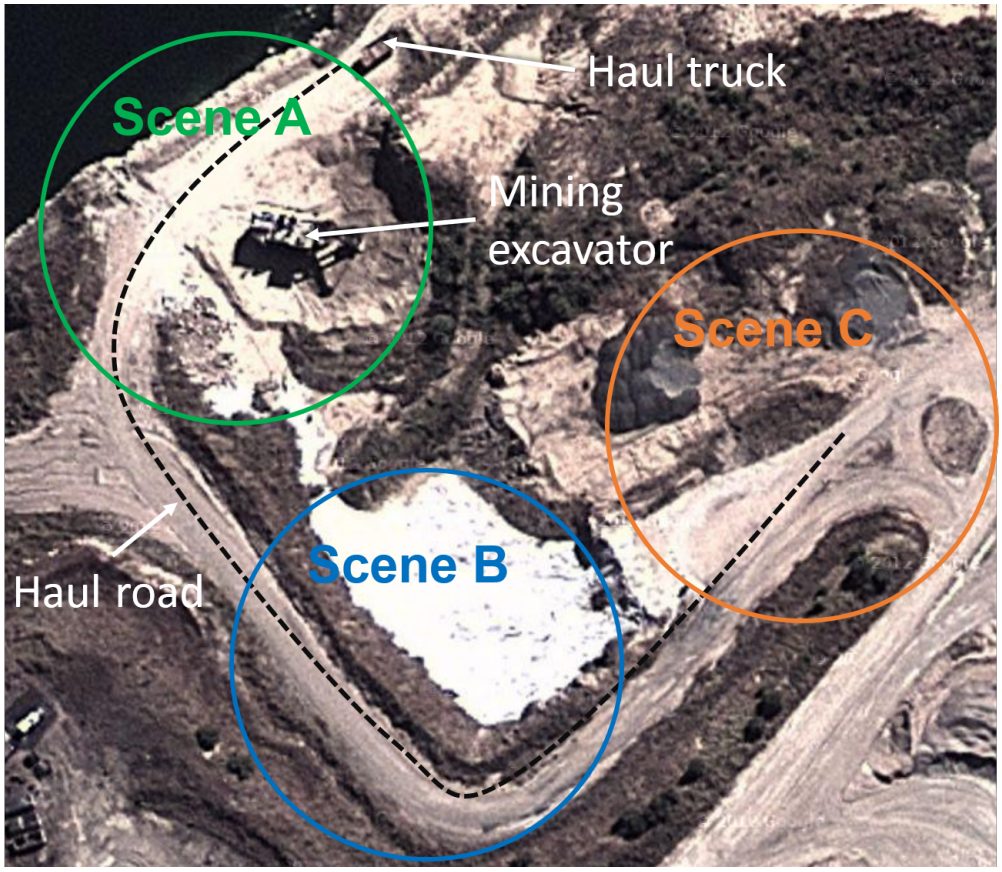
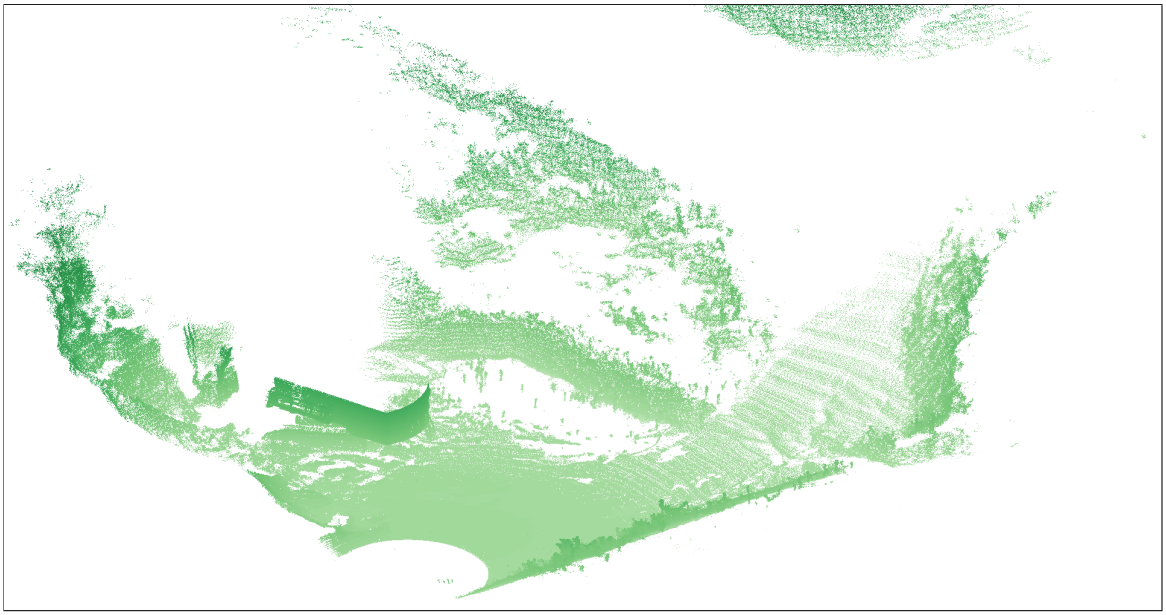


Figure 3.3: The haul road sites (scenes) chosen for this study (Image from Google Inc., 2014).

Scene A: loading area

Scene A is a loading area and includes an excavator digging a working face, see in the right of Fig. 3.4a. The terrain has a low slope with some vegetation and rocks. The scene contains an electric mining shovel, the trailing power cable of the shovel, a pool of water and a berm. The haul truck is visible in the left of Fig. 3.4a.

(a) Scene A: the load area with truck (left) and excavator (right).



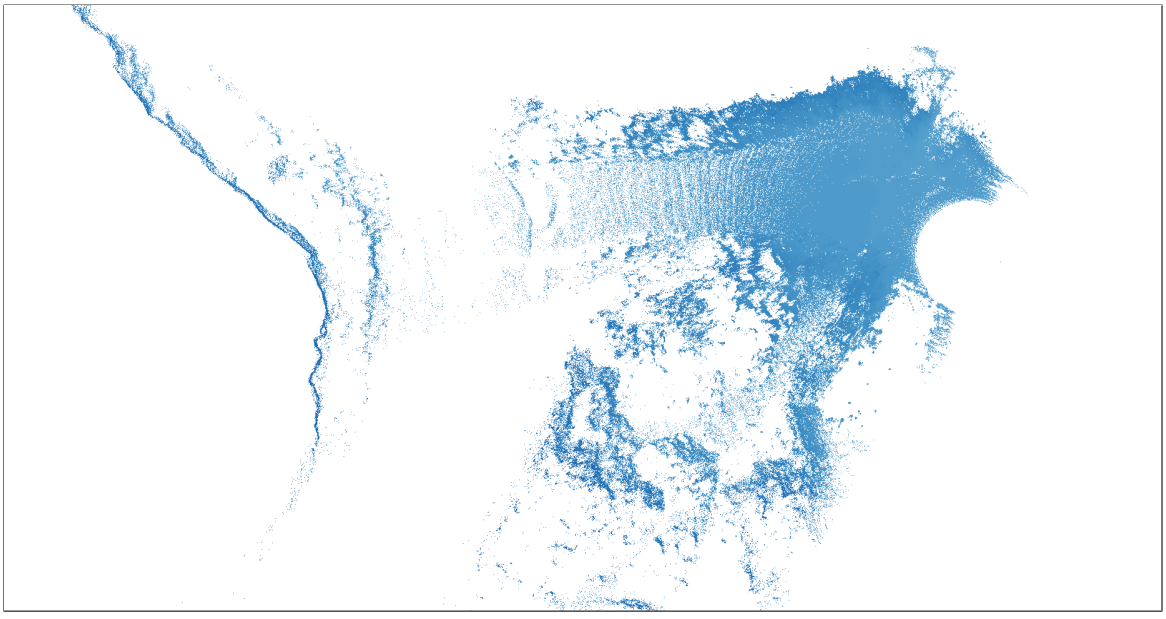
(b) Scene A: Consolidated ground truth point cloud viewed from above-right.

Figure 3.4: Scene A: The loading area.

Scene B: haul road

Scene B is a typical haul road segment, taken from the route between the excavator working scene to the stockpile scene, Fig. 3.5. The truck drives up-hill towards a sheet rock wall approximately 5 m high covered with abundant vegetation. The road is relatively flat with rock and vegetation to the sides of the road.

(a) Scene B: haul road segment.



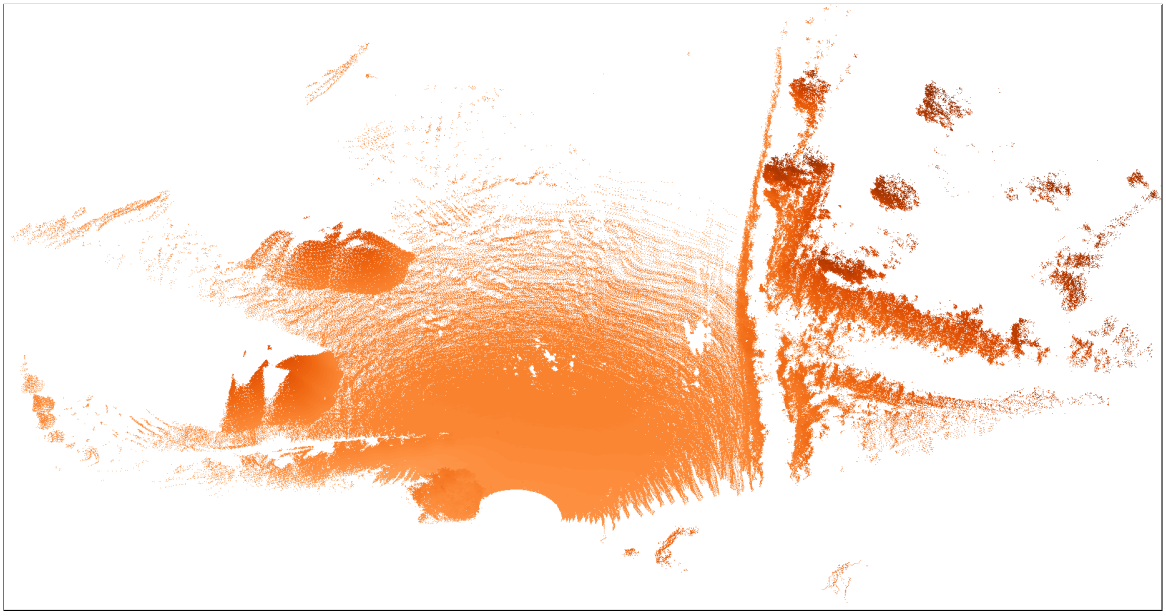
(b) Scene B: Consolidated ground truth point cloud viewed from above.

Figure 3.5: Scene B: haul road.

Scene C: stockpile area

Scene C is a stockpile area, see Fig. 3.6. The terrain is flat and without vegetation. The principal features are two stockpiles of the road-base product produced at the site.

(a) Scene C: the stockpile area. The terrestrial survey station used to evaluate the ground truth is shown in the foreground, but not present in the scan image.



(b) Scene C: Consolidated ground truth point cloud viewed from above.

Figure 3.6: Scene C: stockpile area.

3.2 Metrics of performance

Each ICP variant is tested its ability to construct a map given a sequence of consecutive scans. The map is the consolidation of all points from the sensor into a common frame of reference. Algorithm 2 describes the procedure by how the map is constructed.

Algorithm 2 Mapping procedure

Require: A set of successive scans $\{\mathcal{S}_i\}$ (point clouds). T_0^G is the initial transformation to the global (ground-truth) frame. T_0 is the initial transformation. T_i is the transformation of the i-th scan. $\mathcal{M}_i$ is the i-th scan transformed into the global (map) frame.

Ensure: $\mathcal{M}$ is the map (point could) generated by the union of the transformed scans.

1: $S_0 \leftarrow$ getScan()
2: $\mathrm{T}_0 \leftarrow \mathrm{T}_0^G$
3: $\mathcal{M}_0 \leftarrow \mathrm{T}_0\mathcal{S}_0$
4: $\mathcal{S}_1 \leftarrow$ getScan()
5: **while** $\mathcal{S}_{i+1} \neq \emptyset$ **do**
6: $\quad \mathrm{T}_i \leftarrow$ ICPvariant$(\mathcal{S}_{i-1}, \mathcal{S}_i)$
7: $\quad T_i^G = \mathrm{T}_0 \cdots \mathrm{T}_{i-1}\mathrm{T}_i$
8: $\quad \mathcal{M}_i \leftarrow \mathcal{S}_i(T_i^G)$
9: $\quad \mathcal{M} \leftarrow \bigcup_0^i \mathcal{M}_i$
10: $\quad S_{i+1} \leftarrow$ getScan()
11: **end while**

The performance of an ICP variant is assessed using 100 scans (5 second) to construct a map. ICP performance is quantified by three metrics:

1. Accuracy, $\nu(k)$, quantified by the total RMS error between the consolidated map and the ground-truth, at scan $k = 100$. For each individual scan the RMS error is,

$$E_s(n_k) = \sqrt{\frac{1}{n_k}\sum_{i=1}^{n_k}\left\|\check{\mathbf{p}}_i^k - \tilde{\mathbf{p}}_i^k\right\|^2}, \tag{3.1}$$

where $\check{\mathbf{p}}_i^k$ is the the truth location for the i-th point of the scan k and $\tilde{\mathbf{p}}_i^k$ defines the rotated and translated point of the measure $\mathbf{p}_i$ of the scan k. n_k is the number of points of the i-th scan. A perfect scan match will have $E_s(n_k) = 0$. The total RMS error of the k scans is defined as,

$$\nu(k) = \sqrt{\frac{1}{\sum_{i=1}^k n_i}\sum_{i=1}^k E_s(n_i)^2 n_i}. \tag{3.2}$$

2. Precision, ρ, of the solution found from,

$$\rho(k) = \frac{1}{k}\sum_{i=1}^{k}\Big\|E_s(n_i) - \big(m(i-1) + E_s(1)\big)\Big\|, \tag{3.3}$$

where m is the gradient of the straight line that joins $E_s(1)$ and $E_s(k)$.

3. Computational efficiency, quantified by the relative computation time (rct), that is the time to compute the ICP variant relative to the fastest ICP variant solution.

For illustration, Fig. 3.7 plots each $E_s(n_k)$ for two candidate ICP variants. In relative terms, Variant 2 is accurate but not precise, while Variant 1 is precise but not accurate.

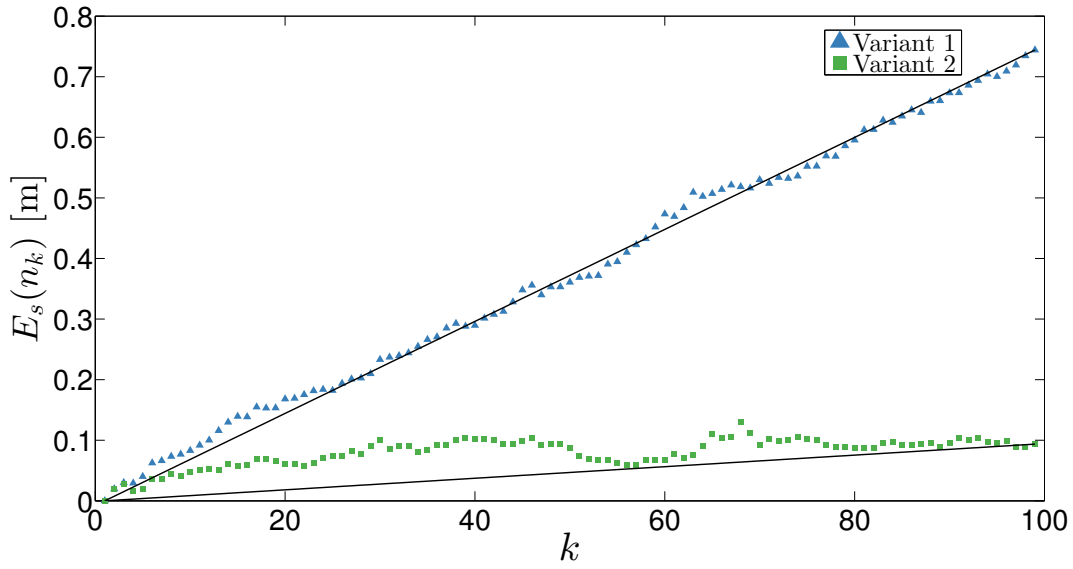


Figure 3.7: RMS error, $E_s(n_k)$, for two ICP variants over 100 scans. k is the scan number.

Tolerance regions for accuracy, precision and computational efficiency are used to grade the ICP variants, based on potential mapping requirements applied to the mining context. ICP variants are considered: accurate if $\nu \leq 0.2\,m$; precise if $\rho \leq 0.1\,m$; and computationally efficient (or fast) if $rct \leq 3$.

3.3 Conclusion

This chapter has described the datasets to be used as a basis for comparing different ICP variants used in scan matching. The data was collected from a mining haul truck platform in an open-pit mining environment. Three metrics of performance were described that capture the accuracy, precision, and computational cost of each variant. The next chapter catalogues published variants of the ICP method and compares their performance against these metrics.

Chapter 4

An evaluation of ICP variants

Chapter 2 outlined the algorithmic foundations of ICP and described in detail three distance metrics used for the minimization step. Chapter 3 described the three scenes to be used for comparing different ICP variants and three metrics against which performance is to be measured. There are many variations on the basic ICP method and the objective of this chapter is to review published methods against the dataset described in Chapter 3 to evaluate their performance.

4.1 Introduction

The ICP methods of Chapter 2 are simple to put into practice and efficient, particularly when implemented with kd-trees for establishing correspondence. However, the application of "vanilla" ICP, as presented in Chapter 2, produces less-than-optimal matching across a range of applications including terrain mapping. Reasons include: i) points from consecutive scans do not map one-to-one; ii) each new scan covers a spatial region different to that in the previous scan; and iii) in the region of overlap, different terrain points are sampled. To improve performance in this and other applications, many variants of ICP have been proposed that curate the raw point cloud data in various ways to improve the match.

In this chapter the performance of different ICP variants is analysed against the reference data sets of Chapter 3, comparing their performance for accuracy, precision, and efficiency with a view towards guiding the selection of scan matching algorithms by ICP. The performance of scan matching algorithms is fundamentally determined by the distribution of points in the point clouds to be matched, see for example Pomerleau et al. (2013). The three scenarios of Chapter 3 are considered typical of such an environment and broadly capture the variation that a scan matching algorithm must be robust to when working with information from such an environment.

There have a been a number of prior studies that compare the performance of different ICP variants. Rusinkiewicz and Levoy (2001) decomposed various adornments and decorations applied to vanilla ICP into a six-stage computational process to examine the convergence speed and accuracy of differ-

ent strategies. The study observes that different variants perform better on different point clouds and recognises the need for deeper insight into the scan matching algorithms. They propose the idea of adaptively choosing variants, depending on point cloud geometry.

Salvi et al. (2007) present a survey of coarse and fine scan matching methods focussing on the accuracy of the resulting model. Methods include: i) the addition of artificial Gaussian noise; ii) varying the number of points used (sampling); and iii) varying the percentage of outliers included. They found that point-to-plane with rejection of paired points (Chen and Medioni, 1992) provided the best performance in term of both accuracy and computational time.

More recently, Pomerleau et al. (2013) presented a framework for comparing different ICP variants and made a comparison of two baseline variants over several data sets taken from different application domains. They show that the performance of the baseline variants vary significantly with different data sets and conclude the need for better ICP variants for natural, unstructured and information-deprived environments. Surface mining environments exhibit these attributes.

The study presented in this chapter is distinguished from these earlier works in three respects: i) its comprehensiveness; ii) the focus of its target application (loosely structured terrain); and iii) its emphasis on how accuracy, precision, and computational efficiency trade-off across different variants.

A total of 20,736 ICP variants are applied to the three data sets. The accuracy of ICP variants is evaluated by comparing the RMS distance error between the last scan in a sequence of one-hundred scans and a ground-truth for that scan, obtained by using an accurate GNSS-IMU navigation system to fuse consecutive scans. Precision is estimated by calculating the deviation of RMS error for each scan from a straight line fit of first and last scan RMS error. Chapter 3 provides a detailed description of the accuracy and precision quantities. The robustness of methods is measured by evaluating performance over three scenes.

The structure of this chapter is as follows: Section 4.2 reviews ICP variants and gives rationale to the methods selected for this study. Comparative results are given in Section 4.3 for all ICP variants. A summary of conclusions is provided in Section 4.4.

4.2 Variants of the iterative closest point algorithm

4.2.1 Computational stages for ICP-based scan matching

Rusinkiewicz and Levoy (2001) consider ICP-based scan matching to consist of six distinct computation stages with the possibility for using different methods or combinations of methods at each stage. The starting point is to adopt this decomposition of scan matching and adapt it to accommodate the new algorithms and ideas which have emerged since this study.

The computational decomposition used in this work is shown in Fig. 4.1. The inputs to the computation are two consecutive scans, where scan $\mathcal{S}_i$ denotes the input cloud, and $\mathcal{S}_{i-1}$ is the reference point cloud to which $\mathcal{S}_i$ is to be matched. Figure 4.1 has six stages:

1. **Point selection:** Data reduction involving the preferential selection a set of points from the input point cloud for scan matching.
2. **Neighbourhood selection:** Establishes a region around each point to determine features associated with the point, *e.g.* normals.
3. **Point matching:** Pair points of the input point cloud to those of the reference point cloud.
4. **Weighting:** Assigns weights to matched point pairs.
5. **Rejection:** Discard point pairs that do not contribute positively to minimization.
6. **Minimization:** The matching of scans by minimization of a metric to bring the input point cloud into alignment with the reference point cloud.

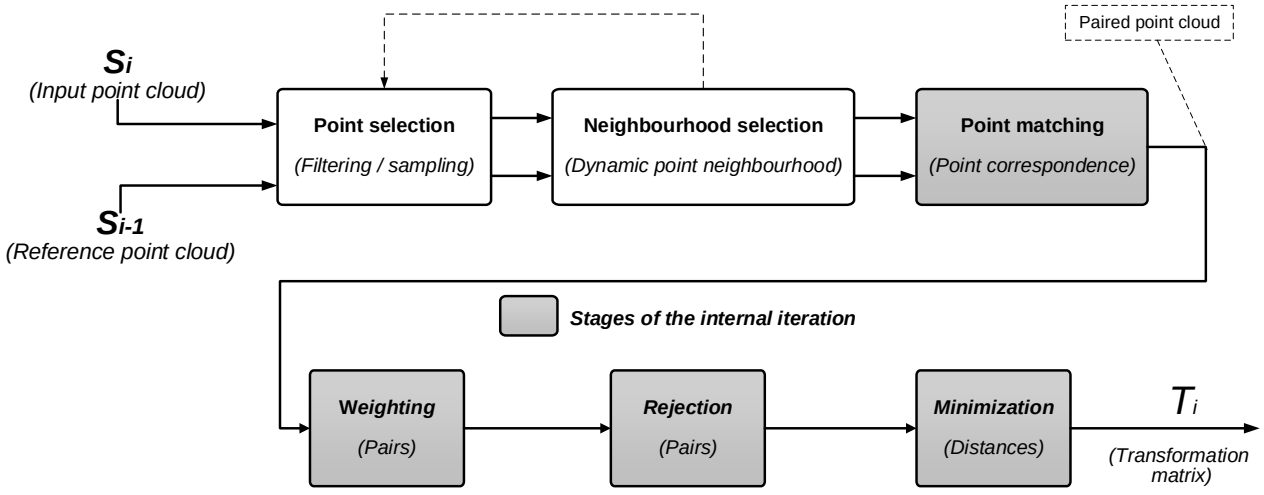
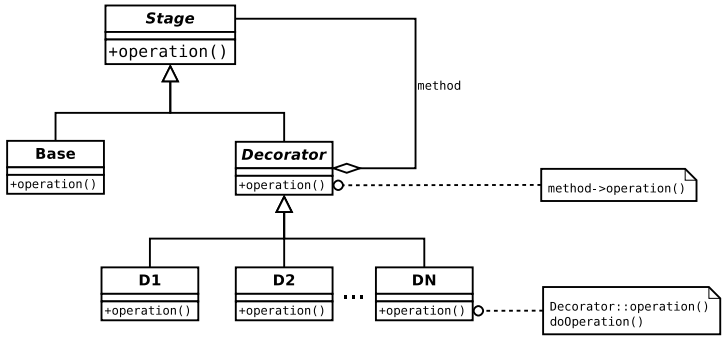
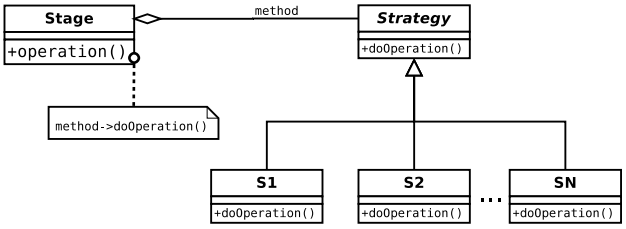


Figure 4.1: The ICP pipeline - a decomposition of the computational stages of the ICP algorithm.

The stages fall into two distinct types. The first type (comprising *point selection* and *rejection*) correspond to stages that filter the input and can be implemented by methods applied alone or in a sequence with each method applying an additional layer of filtering to the data in the manner of a decorator design pattern (Gamma et al., 1995). The second type (comprising *neighbourhood selection*, *matching*, *weighting*, and *minimization*) have the characteristic that they are implemented by one of several alternative algorithms in the manner of a strategy design pattern (Gamma et al., 1995), see Fig. 4.2b.



(a) Decorator pattern.



(b) Strategy pattern.

Figure 4.2: Design patterns used to identify methods of the ICP pipeline.

The adopted methodology in this study is to take a selection of candidate decorations or strategies for each stage in the computation, as appropriate, and explore the performance of various permutations and combinations of the composite algorithms. The large number of potential alternatives at each stage mandates judicious selection of those algorithms to be compared. For reasons of practicality the study is limited to those methods either in common use or, where many alternatives are available, those thought to be in some sense superior.

4.2.2 Point selection

The first stage of the ICP computation process seeks to reduce the input (and reference) point clouds by application of one or more filters. Point selection methods are applied to reduce the number of points used at subsequent stages and to improve the characterization of the underlying data through a meaningful selection of points that encourage fast and accurate convergence to the correct solution. Reducing the number of points provides a practical consideration for timely delivery of an ICP solution, particularly when handling very dense data sets.

It is noted that the points selected for processing at future stages must be chosen judiciously, with there being potential to exclude points useful to the correspondence and minimization stages by poor point selection.

In their seminal paper on ICP, Besl and McKay (1992) proposed that point selection is not needed when there is considerable overlap between point clouds and the number of outliers is not significant. However, for dense point clouds the pragmatic need to reduce the number of points arises in order to

make computation times acceptable. Sub-sampling methods proposed to manage dense point clouds include: random (Masuda and Yokoya, 1995) and uniform (Chen and Medioni, 1992; Zhang, 1994) sampling. However, these basic point selection strategies are generally sub-optimal, deliver slow convergence, and may lead to divergence of the scan matching algorithm. For this study we limit consideration to what might be called discerning algorithms for point selection.

Discerning point selection algorithms typically involve analysis of distinctive attributes or aspects (that is, features) of the point cloud to achieve a well-judged selection of points. It is known that if too many points are chosen from featureless regions, the scan match may converge slowly, converge to a wrong solution (corresponding to a local minima) or, diverge (Gelfand et al., 2003). Divergence is sometimes referred to as instability and one of the more obvious ways in which it arises is when the bulk of points in the clouds to be matched are on planes allowing the solution to "slide around" during alignment.

The normal-space sampling method (Rusinkiewicz and Levoy, 2001) attempts to eliminate solution sliding by selecting a subset of points whose point-normals are as widely and uniformly distributed as possible over the unit sphere. The idea of this method is to reduce translational instability that leads to a better and faster convergence compared with uniform sampling.

Gelfand et al. (2003) extend these ideas, giving prominence to the notion that selected points should constrain the alignment so that meshes converge quickly and accurately during ICP minimization. They termed such point selections geometrically stable. The method estimates the transformations that can cause unstable sliding and selects the points that best constrain these potentially unstable transformations. This is done by defining a six-by-six matrix, C, constructed from k points, $\mathbf{p}_i$, from the input point clouds S_i, and their corresponding normals, $\mathbf{n}_i$, according to

$$C_{6\times 6} = \begin{pmatrix} \mathbf{p}_1 \times \mathbf{n}_1 & \cdots & \mathbf{p}_k \times \mathbf{n}_k \\ \mathbf{n}_1 & \cdots & \mathbf{n}_k \end{pmatrix} \begin{pmatrix} (\mathbf{p}_1 \times \mathbf{n}_1)^T & \mathbf{n}_1^T \\ \vdots & \vdots \\ (\mathbf{p}_k \times \mathbf{n}_k)^T & \mathbf{n}_k^T \end{pmatrix},$$

and selecting the k points so that the condition number of C is as close as possible to one. Only points in the estimated region of overlap between matched scans are selected. Gelfand et al. (2003) argue that the approach produces faster and more accurate scan matching than Rusinkiewicz and Levoy (2001), whilst stabilizing sliding in all spatial dimensions (not just translation). However, Torsello et al. (2011) criticised the method for its tendency to introduce artificial constraints in the presence of noisy point clouds.

Torsello et al. (2011) propose relevance-based sampling, to overcome the introduction of artificial constraints introduced by noise. This approach uses the average local radius of curvature as a distinctiveness measure. A process of integration is used to obtain the measure, improving the sampling robustness to noise. Torsello et al. (2011) present a favourable comparison of the method to uni-

form and normal space sampling, showing convergence to a better fit for different levels of noise. Relevance-based sampling is oriented to computer modelling and is not suitable for large terrain point clouds with non uniform point densities.

Gressin et al. (2012) propose two different point selection methods based on a combination of eigenvalues of the covariance matrix. The first method is based on the dimensionality (linear, planar and scattered) by selecting points with linear behaviours. The second method aims to select points with higher entropy feature values.

Outlier removal via distance threshold has proven to be one of the most used and effective filtering approaches. The method seeks to remove points that do not contain a nearest neighbour in a specified distance threshold. A similar approach is filtering points with density lower than a given threshold (Lalonde et al., 2006), such that a constant density is maintained throughout a point cloud. Other point selection methods include the use of surfaces such as the kernel density filter (Schall et al., 2005) which is based on normals and eigenvalues of the covariance matrix but using kernels to characterize the surface.

A decorator pattern is used to provide the computational framework for implementing point selection, allowing different combinations of selection methods to be explored. The order in which methods are applied is based on a simple heuristic. Featureless methods are placed first because they don't need interest point detection methods. Specifically, the *outlier removal* method is applied first to reduce the outliers in the point cloud, followed by *density filter* method that evaluates the density distribution throughout the point cloud, then the point cloud is sampled by the *geometrical stable sampling* method and finally the *entropy feature* filter or *dimensionality selector* methods one applied to what remains. In total, 24 combinations of point selection methods are considered.

Table 4.1 overviews various methods used for point selection and identifies those to be considered in this study.

Table 4.1: Point selection methods.

Decoration	**Reference**	**Included**	**Rationale**
All points	Besl and McKay (1992)	Yes	Baseline.
Range sampling	Obvious heuristic, widely used but not explicitly referenced.	No	Heuristic method without underlying theoretical basis.

Continued on next page

Table 4.1 – *Continued from previous page*

Decoration	**Reference**	**Included**	**Reasons**
Random sampling	Masuda and Yokoya (1995)	No	Used to decimate dense point clouds and shown to be inferior to the entropy feature filter and dimensionality based selection in Gressin et al. (2012).
Uniform sampling	Chen and Medioni (1992); Zhang (1994)	No	Used primarily to decimate point clouds and shown in Gelfand et al. (2003) to be inferior to Geometrically stable sampling.
Normal-space sampling	Rusinkiewicz and Levoy (2001)	No	Shown in Gelfand et al. (2003) to be inferior to geometrically stable sampling.
Geometrically stable sampling	Gelfand et al. (2003)	Yes	Sampling method designed to select points that constrain alignment.
Relevance space sampling	Torsello et al. (2011)	No	Method is targeted to well-defined, smooth geometries and primarily targets the avoidance of artificial constraints that might arise in geometrically stable sampling due to measurement noise. Not appropriate to environments studied in this work.
Entropy feature filter	Gressin et al. (2012)	Yes	Known to give a meaningful selection of points.
Dimensionality based selection	Gressin et al. (2012)	Yes	Keeps points with planar behaviour.
Outliers removal	Xie et al. (2004)	Yes	Simple and commonly used algorithm to remove spurious points.
Density filter	Lalonde et al. (2006)	Yes	Useful algorithm to bound the point density.

Continued on next page

Table 4.1 – *Continued from previous page*

Decoration	**Reference**	**Included**	**Reasons**
Kernel density filter	Schall et al. (2005)	No	Method oriented to clustering by moving each point to the most likely position into a cluster. This process is very slow and modifies the original data.

4.2.3 Neighbourhood selection

An accurate representation of point cloud geometry influences both the matching and minimization stages of the registration process. Neighbourhood selection is used to determine the set of points necessary to accurately describe the underlying surface geometry, through, for example, the calculation of surface normals. Selected neighbourhoods are used in *point selection* and for calculation of features.

The simplest approach to neighbourhood selection is to use a fixed radius or number of points to select the neighbourhood for each point in the point cloud. However, when the density of points varies significantly with the distance to the sensor origin, the selection of the neighbourhood size requires a trade-off in the representation of near and far features. A small neighbourhood may not provide a suitable representation of geometry distant from the sensor origin, and a large neighbourhood may distort the representation of geometry close to the sensor.

To mitigate the effects of point density variation the neighbouring points can be dynamically selected based on a local metric. Lalonde et al. (2005) suggest choosing a neighbourhood size of radius, r, that minimizes the expected angular deviation of the computed normal of a point from its true normal. The approach builds on the work of Mitra et al. (2004) for estimating normals in a noisy point cloud. The basic idea is that the optimal normal vector is bounded by an expression depending on the noise of the point cloud, the curvature of the underlying manifold, the density of points, and the neighbourhood size. Eqn. 4.1 shows the optimal radius obtained by minimizing the expression of the bounded normal vector against the neighbourhood radius, where r^* is the optimal radius, κ is the curvature, σ_n^2 is the variance of the sensor noise, ρ is the local point density, ϵ is a small constant, c_1 and c_2 are constants which depend on the distribution of the point cloud.

$$r^* = \left(\frac{1}{\kappa} \left(c_1 \frac{\sigma_n}{\sqrt{\epsilon \rho}} + c_2 \sigma_n^2 \right) \right)^{\frac{1}{3}} \tag{4.1}$$

Curvature and density are obtained within an initial radius or neighbourhood, and the resulting radius

is used to find a new neighbourhood. This process is repeated for a fixed number of iterations. A complete derivation of the method is found in Mitra et al. (2004).

The method of Mitra et al. (2004) and Lalonde et al. (2005) are included as candidate neighbourhood selection methods under the name of "bounded radius".

Point geometry is important to determining the size of the point neighbourhood. The extent of the neighbourhood has to be chosen to appropriately preserve one geometrical feature over others. Taking that into consideration, Demantké et al. (2011) address problem of the dynamic neighbourhood selection by evaluating the entropy feature over a varying radius and selecting the radius that minimizes the entropy feature. A low entropy feature is indicative of a dominant dimensionality, e.g. linear, planar or scattered. The selection of the optimal neighbourhood is chosen to emphasize one of the geometrical dimensions.

$$r^* = \underset{r \in [r_{min},\, r_{max}]}{\arg\min} \; E_f(r), \tag{4.2}$$

r_{min} and r_{max} are sampled using a quadratic distribution whereby more sampling occurs close to the minimum bound.

Wiemann et al. (2010) propose a simple approach to cope with low density regions by taking into account the shape of bounding boxes enclosing a neighbourhood. If the shape is elongated, with insufficient points, the neighbourhood has to be enlarged until the bounding box is square like.

Neighbourhood selection is described by a strategy pattern whereby one of several alternatives is used. Table 4.2 summarize the various methods available and identifies those considered in this study.

Table 4.2: Neighbourhood selection.

Strategy	**Reference**	**Included**	**Reasons**
Fixed neighbourhood (or radius)	Obvious heuristic, widely used but not explicitly referenced.	Yes	Baseline.
Bounded radius for sampled surfaces	Lalonde et al. (2005)	Yes	Strong theoretical reasoning with good empirical results.

Continued on next page

Table 4.2 – *Continued from previous page*

Strategy	Reference	Included	Reasons
Bounded radius for sampled curves	Unnikrishnan et al. (2006)	No	Method focussed on curves that is not suitable for the problem of this work.
Entropy feature minimization	Demantké et al. (2011)	Yes	Based on the entropy feature of the neighbourhood.
Density adaptation	Wiemann et al. (2010)	Yes	Simple method that aims to maintain a square shaped bounding box for neighbouring.

4.2.4 Point matching

Point matching produces a correspondence between points from the input point cloud, $\mathcal{P}^n$, and the reference point cloud, $\mathcal{Q}^m$. The output of the process is a pairing of points that is used as the basis for minimizing the misalignment between point clouds.

Correspondence is achieved by finding the closest point in the input cloud to each point in the reference point cloud. There is no explicit requirement for uniqueness in the correspondence relationships, however, iteration of the ICP algorithm should increase the number of unique closest points. The definition of "closest point" is the defining characteristic of the scan matching method. Besl and McKay prescribe a closest point strategy based on the Euclidean distance between points, which remains the principal method applied to fine registration.

To complete an exhaustive search between all point of $\mathcal{P}^n$ into $\mathcal{Q}^m$ attracts a computational cost of $\mathcal{O}(\mathrm{nm})$. Nearest-neighbour-search (NNS) methods are used to reduce the computational load associated with point matching. Elseberg et al. (2012) provide a comparison of NNS strategies applied to scan matching, reporting a preponderance of kd-tree structure with a kNN search technique. For this reason the kd-tree is applied in this work to find nearest neighbours.

Heuristics that constrain pair correspondence are applied to improve the robustness of the NNS by reducing the matching of unrelated points. Pulli (1999) proposes a constraint based on the difference between normal angles, only allowing the matching if the difference of normal angles from both points is less than 45 degrees. A similar approach is used by Godin et al. (1994) to match points

only if their intensity compatibility is greater than a given threshold. The application of constrained correspondence methods is similar to the pair rejection methods discussed in Section 4.2.6.

Further enhancement of the nearest neighbour search is made possible by using low dimension descriptors of the point cloud geometry as a basis for point matching. The implementation is typically realized through the definition of an enhanced distance measure, $d(p_i, q_j)$, that is a weighted function of the properties attributed to the two points, p_i and q_j,

$$d(p_i, q_j) = \alpha d_e(p_i, q_j) + \beta d_{f_1}(p_i, q_j) + \gamma d_{f_2}(p_i, q_j), \tag{4.3}$$

where d is the enhanced distance, d_e is the Euclidean distance from a point p_i, of the input, to a point q_j on the reference, d_{f_1} and d_{f_2} are feature distances associated with p_i and q_j, and α, β and γ are weights applied to d_e, d_{f_1} and d_{f_2}, respectively. The weightings are usually set empirically through trial and error.

Feldmar et al. (1995) present a similarly formatted enhancement to the nearest neighbour search with the application of normal vectors. With their approach the search space for point matching grows from three to six dimensions. Contributions from Sharp et al. (2002), Schutz et al. (1998), and Akca (2005) cover the application of curvature, moment invariants, spherical harmonic invariants, colour, and intensity as descriptors applied to the matching process.

An alternative to using the distance across points is to use high dimensionally features of the point cloud geometry to drive point correspondence (Peng, 2012). Such methods are often applied to coarse registration due to the robustness to a large initial misalignment between point clouds. Their application to fine registration is not explored in this thesis.

Table 4.3 summarizes the matching methods considered in this study. The computational framework for applying point matching follows a strategy design pattern where the family of methods can be applied interchangeably to the ICP algorithm.

Table 4.3: Point matching methods.

Strategy	**Reference**	**Included**	**Reasons**
Nearest neighbour (NN)	Besl and McKay (1992)	Yes	Baseline.
NN constrained by normal angle	Pulli (1999)	No	Simple heuristic analogues to rejection by normal deviation but less restrictive.
NN constrained by intensity	Godin et al. (1994)	No	This work is focussed on geometrical relations only, excluding colour and intensity information.

Continued on next page

Table 4.3 – *Continued from previous page*

Strategy	**Reference**	**Included**	**Reasons**
Coarse matching	Peng (2012)	No	Coarse registration method are not revised in this work.
NN enhanced by normals	Feldmar et al. (1995)	Yes	Normals are very good describers of the underlying surfaces.
NN enhanced by colours	Schutz et al. (1998)	No	This work is focussed on geometric relations only, excluding colour and intensity information.
NN enhanced by moment invariants	Sharp et al. (2002)	Yes	Moment invariants have been used extensively in image processing.
NN enhanced by curvature	Sharp et al. (2002)	No	Very similar to NN enhanced by normals and shows a slightly worse performance than moment invariants.
NN enhanced by spherical harmonics	Sharp et al. (2002)	No	Complicated feature based on a spherical representation of the data.

4.2.5 Weighting

The weighting of matched pairs uses local or global contextual information to modify the distance function associated with each pair. The intent is to influence the individual contribution of matched pairs to the minimization process thereby improving the ICP performance.

Weighting by distance (Godin et al., 1994) assigns lower weights to pairs with greater separation. Another approach is to weight according to the scalar product of the normal vectors associated with the paired points, reducing the contribution of those pairs with disparate normals (Rusinkiewicz and Levoy, 2001). These strategies are applied in this study along with the baseline of constant weighting applied to all matched pairs. Given previous studies (Rusinkiewicz and Levoy, 2001; Gressin et al., 2012), the expectation is that pair weighting will not be a significant factor in the performance of ICP variants.

Khoshelhama et al. (2013) propose a weighting method based on the variance in the depth axis of an image given by a Kinect sensor. This approach is not suitable for the LiDAR data of this study as the

approach requires the variance to be equal in all axes.

Table 4.4 provides an overview of the methods used for weighting. As separate strategies they are applied interchangeably to the ICP algorithm.

Table 4.4: Weighting methods.

Strategy	Reference	Included	Reasons
No weighting (constant of 1)	Besl and McKay (1992)	Yes	Baseline.
Weighting by distance	Godin et al. (1994)	Yes	Classic method for point-to-point minimization.
Weighting by normal angular deviation	Rusinkiewicz and Levoy (2001)	Yes	Direct relation with minimization by point-to-plane distance.
Weighting by axes variance	Khoshelhama et al. (2013)	No	Not suitable for LiDAR data.

4.2.6 Rejection

Pair rejection extends the pair weighting operation by discarding those pairs that disrupt the minimization of the distance function. This process seeks to improve the convergence of the algorithm by eliminating pairs representing "false positives", points without overlap, or point pairs that are outliers. Besl and McKay (1992) did not include the rejection of matched pairs as a formal stage of the ICP algorithm. They did however identify that outliers and occlusions negatively impact the performance of ICP and identified the mitigation of their affect as an area of future work.

The Euclidean distance of the paired points provides a simple and powerful way to identify outliers or occluded points (Zhang, 1994). The simplest approach is to use a fixed threshold distance, across all points and all iterations, as a basis for rejecting points. This approach has significant limitations due to variations in point cloud geometry, poor robustness to different point matching scenarios, and the global reduction in average distance as the solution converges.

Zhang (1994) proposed an adaptive distance threshold based on the mean and variance of the distances between pairs. In their approach a target optimal average error is used to dynamically calculate a distance threshold as a function of the mean and variance across all point pairs. An alternative formulation (Rusinkiewicz and Levoy (2001); Pulli (1999)) applies a threshold percentage to identify

the worst pairs ordered by distance. The threshold percentage approach is robust, but the appropriate setting of the threshold percentage is dependent on the type of registration problem. For the fine registration example problem used in this study, a 10% threshold for pair rejection was applied to all scenes, determined through trial and error.

Point normal comparison (Rusinkiewicz and Levoy, 2001) provides a similar rejection method as the threshold distance. A large difference in the normals of matched points suggest that points do not share the same local geometry, providing a means for rejection. In this study, a fixed 5° degree angle threshold is set.

Table 4.5 overviews the methods used for pair rejection and identifies those to be considered in this study. The computational framework follows a decorator pattern allowing individual methods to be used in combination. If rejection by distance and normal are used in combination, rejection by distance is applied first to provide an "initial cut".

Table 4.5: Pair rejection methods.

Decoration	**Reference**	**Included**	**Reasons**
No rejection	Besl and McKay (1992)	Yes	Baseline.
Rejection by distance threshold	Zhang (1994)	No	The threshold depends on the geometry and outliers of the point cloud. A fixed threshold for each scenario and ICP variant is impossible.
Rejection of a fixed percentage of pairs with worst distance	Rusinkiewicz and Levoy (2001)	Yes	Overcomes the problem of a fixed threshold in a simple and robust manner.
Rejection by angular deviation	Rusinkiewicz and Levoy (2001)	Yes	Very robust rejection with direct benefits in the minimization of point-to-plane distance.
Rejection by an adaptive distance threshold based on the mean and variance of pair distance	Zhang (1994)	Yes	Semi-autonomous threshold, depending on the expected average distance error.

4.2.7 Minimization

Methods for minimization have been covered in Chapter 2. The three methods for minimization considered in this study are summarized in Table 4.6.

Table 4.6: Minimization methods.

Strategy	Reference	Included	Reasons
Point-to-point	Besl and McKay (1992)	Yes	Original method.
Point-to-plane	Chen and Medioni (1992)	Yes	Baseline.
Generalized	Segal et al. (2009)	Yes	Generalization of point-to-point and point-to-plane minimization.

4.2.8 Methods selected for evaluation

Table 4.7 summarizes the variants to be compared. ICP variants are identified by an alphanumeric code where each letter or number represents a different method. The code de3z1α1, for example, employs: (i) geometrically stable sampling (code d) and entropy feature filtering (code e) for point selection; (ii) bounded radius (code 4) for neighbourhood selection; (iii) nearest neighbour enhanced by normals (code z) for point matching; (iv) distance (code 1) weighting; (v) no rejection (code α); and (ii) point-to-plane ICP (code 1) for minimization.

In order to determine the total number of ICP variants given by the methods presented in Table 4.7, it has to be considered that the "Entropy feature filter" and "Dimensionality based selection" are mutually exclusive. Point selection and rejection methods can be applied combinatorially (except for "All points" and "No rejection" methods). The following equations show the calculation of the total number of ICP variants evaluated in this chapter. Where N_{Stage} denotes the number of variants or combinations in each stage.

$$
\begin{aligned}
N_{\text{Point selection}} &= \binom{6}{1} + \left(2\binom{4}{2} - \binom{3}{2}\right) + \left(2\binom{4}{3} - \binom{3}{3}\right) + 2\binom{4}{4} = 24, \\
N_{\text{Neighbourhood selection}} &= 4, \\
N_{\text{Matching}} &= 3, \\
N_{\text{Weighting}} &= 3, \\
N_{\text{Rejection}} &= \binom{4}{1} + \binom{3}{2} + \binom{3}{3} = 8, \\
N_{\text{Minimization}} &= 3, \\
N_{\text{Point selection}} \times N_{\text{Neighbourhood selection}} \times N_{\text{Matching}} \times N_{\text{Weighting}} \times N_{\text{Rejection}} \times N_{\text{Minimization}} &= 20,736, \\
\text{ICP variants} &= 20,736.
\end{aligned}
$$

Table 4.7: Selected strategies for performance comparison.

Stage	**Code**	**Strategy**
Points selection	a	All points
	b	Outliers removal filter
	c	Density filter
	d	Geometrically stable sampling
	e	Entropy feature filter
	f	Dimensionality based selection
Neighbourhood selection	1	Constant
	2	Entropy feature minimization
	3	Density adaptation
	4	Bounded radius
Point matching	x	Nearest neighbours (NN)
	y	NN enhanced by moment invariants
	z	NN enhanced by normals
Weighting	0	Constant
	1	Distance
	2	Normals compatibility
Rejection	α	No rejection
	β	Distance by worst percentage
	γ	Angular deviation
	δ	Adaptive distance by variance
Minimization	0	Point-to-point

Continued on next page

Table 4.7 – *Continued from previous page*

Stage	Code	Strategy
	1	Point-to-plane
	2	Generalized

4.2.9 Implementation

The methods selected for evaluation have been implemented in a bespoke software framework written in the C++ language for the purpose of comparative evaluation. All methods were written from scratch making reference to the original descriptions of them. To minimize the effects of result bias due to implementation, where possible methods have been reviewed against and compared in performance with various public domain codes. These codes include Point Cloud Library, PCL (Rusu and Cousins, 2011), an open-source library of algorithms for point cloud processing, the 3D Toolkit, 3DTK (Nuechter et al., 2012), an open-source library specialized in 6D-SLAM and the Robot Operation System, ROS (Quigley et al., 2009), open-source libraries for creating robot applications.

Various methods require normal vectors to be computed. Normal vectors are a geometric property of surfaces discretized by point clouds and each point in a cloud can be considered to have an associated normal that is established from neighbouring points. Numerous approaches have been proposed for determining normal vectors. For example, the use of k nearest neighbours to fit a tangent plane a point Hoffman and Jain (1987); Huang and Menq (2001) or a local quadric surface of similar Yang and Lee (1999). Klasing et al. (2009) evaluate several methods to estimate normal vectors on a point neighbourhood, finding that principal component analysis (PCA) is superior to alternative methods in performance and speed. The PCA approach is selected accordingly as the method used to calculate normals where they are required.

4.3 Results and observations

This section presents ten observations based on 62,208 ICP-variant/scene combinations, comprising 20,736 distinct variants applied to the three candidate scenes (data sets). The data for each scene comprises 100 consecutive point cloud scans made at 20 Hz over a 5 second interval.

Summary of observations

Observation 1: *The performance of the scan match as measured by accuracy, precision and computation efficiency varies significantly across the ICP variants making judicious selection an imperative where a minimum level of performance must be met.*

Observation 2: *The performance of ICP variants depends on the scene.*

Observation 3: *Computation efficiency for ICP variants is not correlated with accuracy or precision.*

Observation 4: *There is no single best ICP variant as measured by accuracy, precision, and computation efficiency.*

Observation 5: *Minimization is best performed by point-to-plane ICP and is preferred over generalized ICP and point-to-point ICP.*

Observation 6: *There is a preference to match points using the nearest neighbour enhanced by normals based on this method being the most prevalent and the most accurate ICP variants.*

Observation 7: *Pair rejection improves scan matching. Rejection by angular deviation can be used effectively in combination with rejection by adaptive distance and rejection by worst distance percentage.*

Observation 8: *Bounded radius is the preferred method for neighbourhood selection.*

Observation 9: *There is no clear preference among point selection methods although outlier removal, dimensionality based selection and geometrically stable sampling appear commonly in the preferred variants.*

Observation 10: *There is no clear preference among weighting methods and this stage can be reasonably removed from the computation process.*

Evidence for observations

Scatter plots of accuracy-precision and efficiency-accuracy for the ICP variant/scene combinations are shown in Fig. 4.3 and Fig. 4.4, respectively. Data is coloured by scene. In general it should be expected that good accuracy implies high precision and this is apparent in the scatter. The converse also holds, that is bad accuracy implies low precision. Overall, a significant spread in performance against the metrics is apparent and the following general observations can be made.

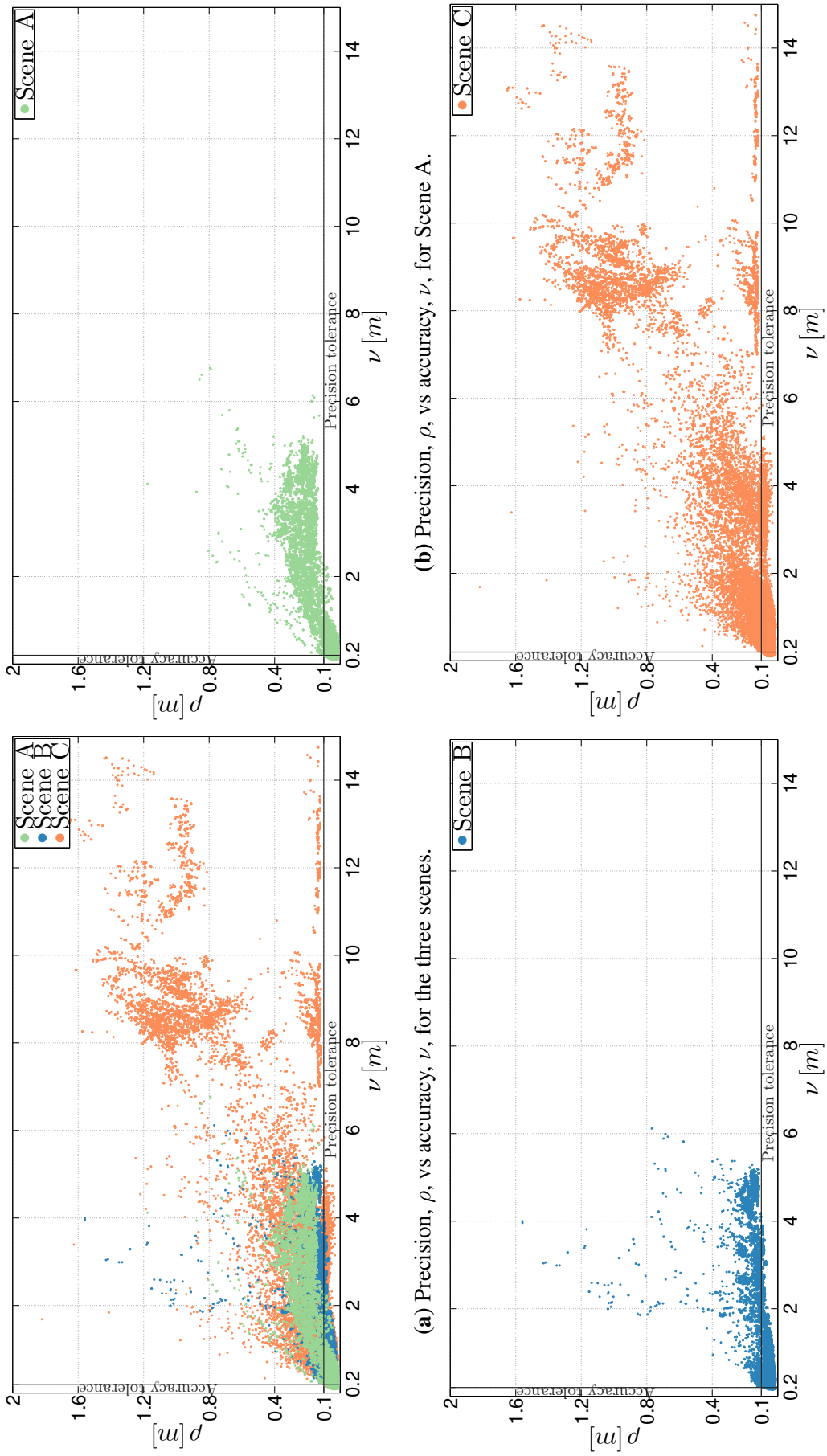
Observation 1: *The performance of the scan match as measured by accuracy, precision and computation efficiency varies significantly across the ICP variants making judicious selection an imperative where a minimum level of performance must be met.*

Observation 2: *The performance of ICP variants depends on the scene.*

These observations are echoed in other works. Observation 1 can be said to account for the large number of ICP variants that have been proposed while Observation 2 is consistent with Pomerleau et al. (2013) who showed the performance of ICP variants differ significantly across different data

sets. These observations hit at a key issue in the use of ICP: each data set (scene) seemingly has its own best variant for scan matching. This is an important point and it focusses the challenge towards identifying those combinations of methods that generally perform well across different scenes.

It is insightful to understand why, across all ICP variants, Scenes A and B have tighter accuracy and precision than Scene C, see Figs. 3.4, 3.5 and 3.6 to refer to the scenes. Observe that Scene C has a planar ground plane, consequently, scan matching for this scene are prone to sliding. This is believed to be the cause of the significant variation observed, and in particular, accounts for the large number of outliers. Scene A exhibits the best overall performance. The excavator present in this scene provides two large flat and orthogonal regions in the point cloud that become effective features for accurate scan matching. Scene B performs well generally but has fewer accurate and precise variants when compared with Scene A. Scene B is characterized by significant vegetation (scattered distribution of points) to the sides of the road way and a significant rock wall that is irregular in form and on which the calculation of point normal is sensitive resulting in local minima.



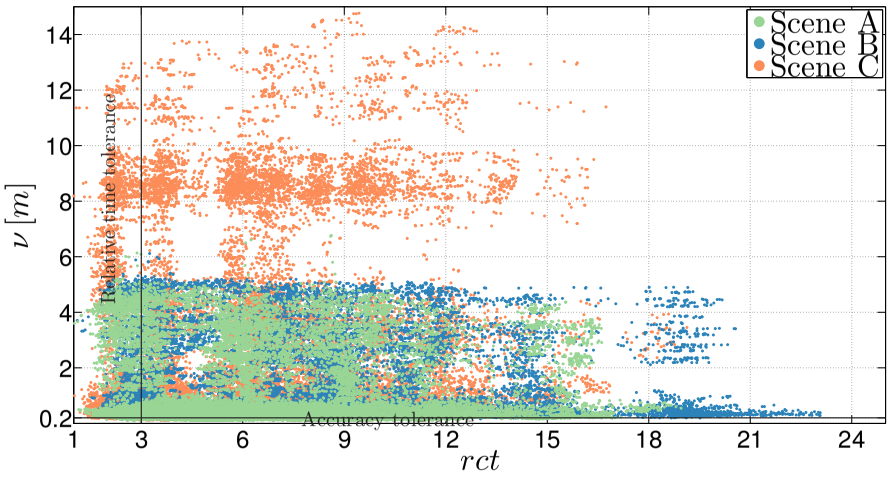
(a) Precision, ρ, vs accuracy, ν, for the three scenes.

(b) Precision, ρ, vs accuracy, ν, for Scene A.

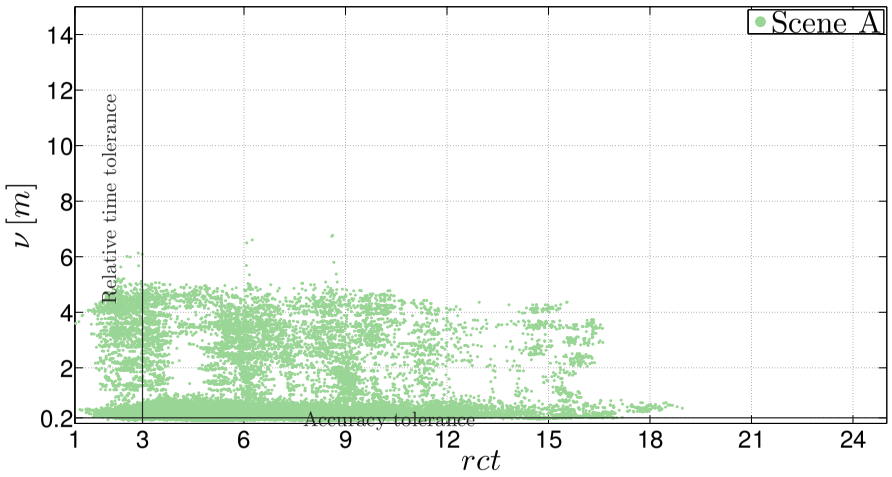
(c) Precision, ρ, vs accuracy, ν, for Scene B.

(d) Precision, ρ, vs accuracy, ν, for Scene C.

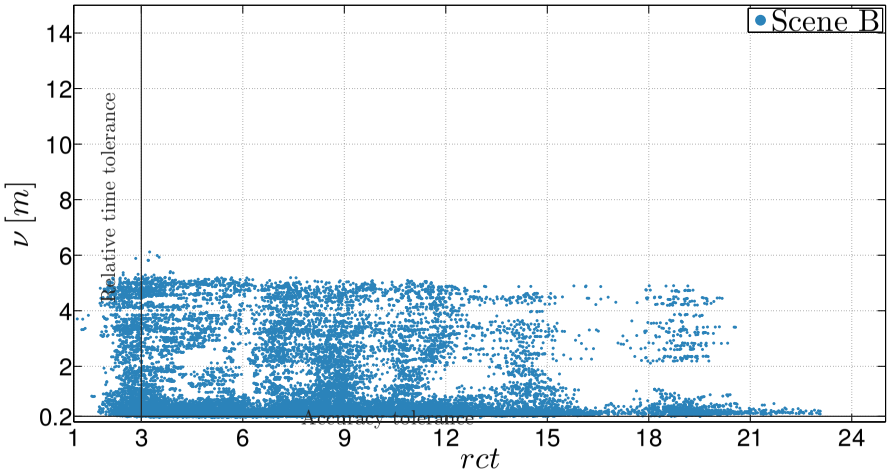
Figure 4.3: Precision vs accuracy of 20,736 candidate ICP variants for the three representive scenes.



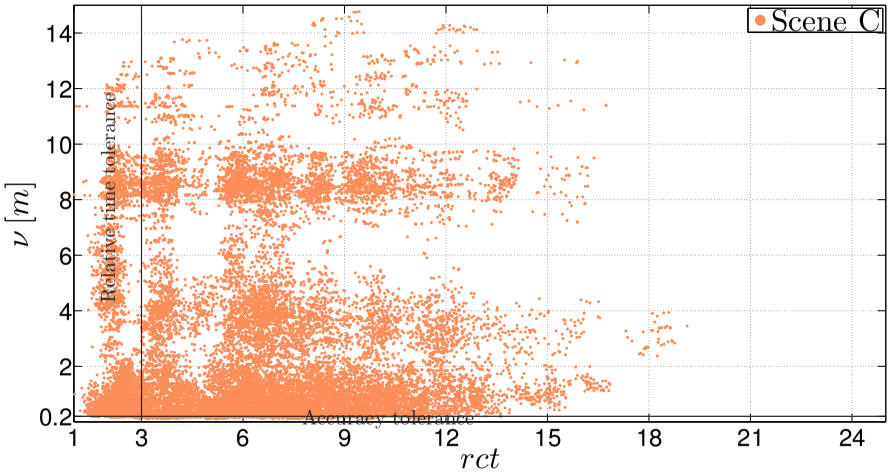
(a) Accuracy, ν, vs computational efficiency, rct, for the three scenes.



(b) Accuracy, ν, vs computational efficiency, rct, for Scene A.

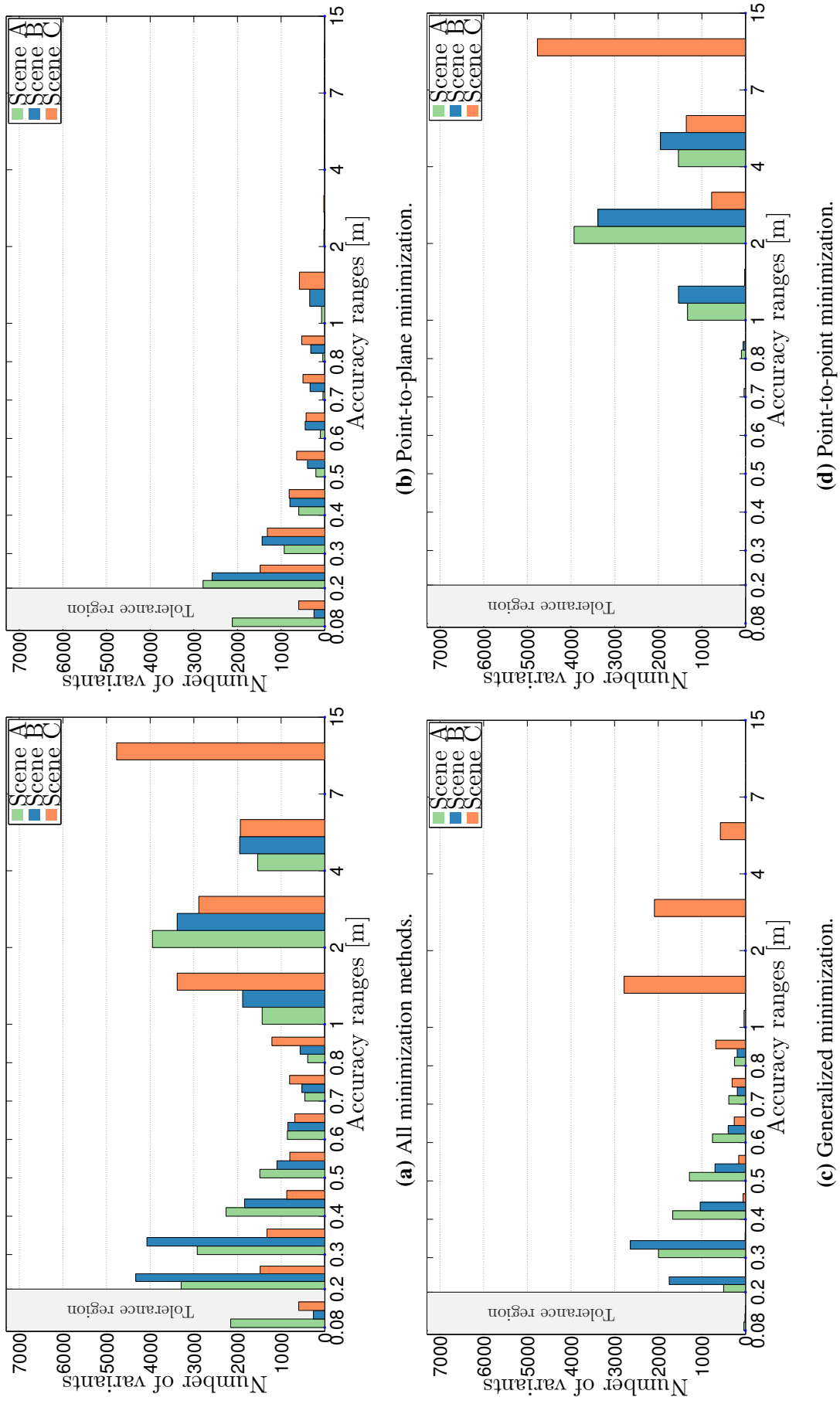


(c) Accuracy, ν, vs computational efficiency, rct, for Scene B.



(d) Accuracy, ν, vs computational efficiency, rct, for Scene C.

Figure 4.4: Accuracy vs computational time of 20,736 candidate ICP variants for the three representive scenes.



(a) All minimization methods.

(b) Point-to-plane minimization.

(c) Generalized minimization.

(d) Point-to-point minimization.

Figure 4.5: Accuracy histograms for the scenes, organized by minimization method.

Observation 3: *Computation efficiency for ICP variants is not correlated with accuracy or precision.*

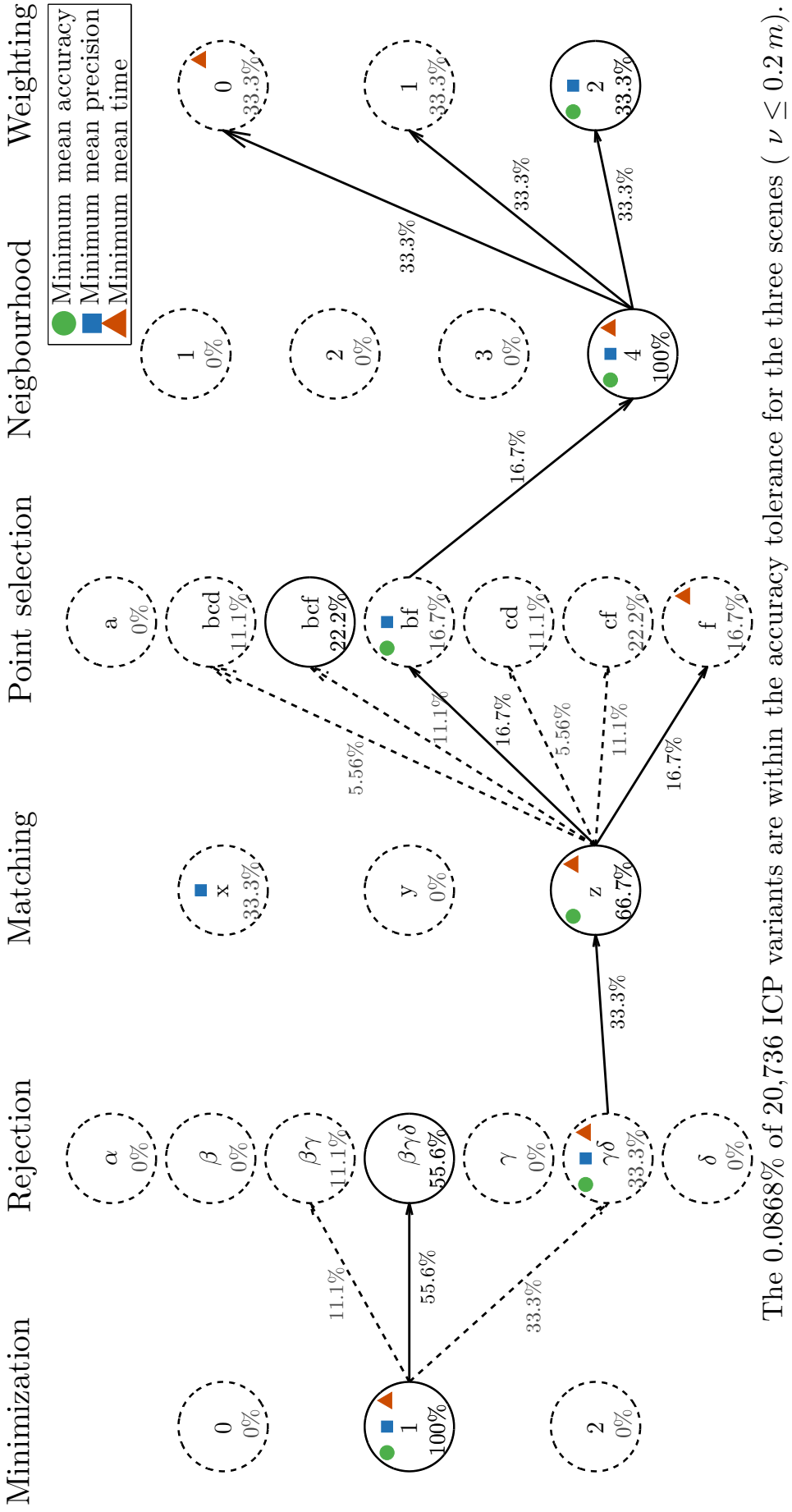
Observation 3 runs contrary to the general expectation that accuracy and precision should improve with the amount of computation effort invested in the scan match. Figure 4.4a, in particular supports the view that computation efficiency is not traded against accuracy, although it is to be expected that some combinations of methods will general perform more accurately, more precisely and have lower computation efficiency than others. It is notable that Scene B has a large number of variants with computation efficiency, rct, greater than 18 whereas for Scene A and C almost all variants have rct less than 15 times the fastest variant. The flat ground plane in Scene C causes solutions to converge to local minima and this dually results in fewer iterations for convergence of the minimization step and a less accurate scan match.

The challenge is to identify those combinations of methods that are best performed across the three scenes in anticipation of these being well performed for scenes generally. The selection of the variants judged to be best is made by thresholding against the metrics of performance in the three scenes together. Three sets of variants are identified: (i) the set of accurate variants; (ii) the set of precise variants; and (iii) the set of fast variants.

The set of accurate variants comprise those with accuracy of less that 0.2 m; 0.0868 % of the 20,736 ICP variants are considered accurate. Precise variants are those whose precision measure across the three scenes is less than 0.1 m; 35.9 % fall below the precision threshold. Computationally efficient variants comprise those whose relative computation time is less than 3; 6.08 % of variants fall below the relative computation time threshold. None of the variants meet all of the threshold conditions. Figures 4.6, 4.7, and 4.8 are visual depictions of the composition of these accuracy, precision and computational efficiency sets by method.

Taking Fig. 4.6 as an example, these graphs should be read as follows. All of the most accurate variants use Method 1 (point-to-plane) for minimization. 33.3 % use the combined Method $\gamma\delta$ (γ, angular deviation; δ, adaptive distance) for rejection. The most accurate, precise and computational efficient solution uses Method $\gamma\delta$, and this is indicated by the solid green circle, the solid blue square and the solid red triangle, respectively. Similarly, 66.7 % of the variants use Method z (nearest neighbour enhanced by normals) for point matching, with the most accurate variants in this set. 33.3 % of the variants use Method $\gamma\delta$ for rejection with Method z for matching. And so the diagram continues with 16.7 % of ICP variants using the combination of Methods b (outlier removal) and f (dimensionality selection) for point selection. All three weighting methods are used with the same 33.3 %. The most accurate variant, on average, has the alpha numeric code: bf4z0$\gamma\delta$1.

An examination of the performance of ICP variants against these data sets reveals several observations that are detailed below.



The 0.0868% of 20,736 ICP variants are within the accuracy tolerance for the three scenes ($\nu \leq 0.2\,m$).

Figure 4.6: Method relationship diagram within the accuracy tolerance.

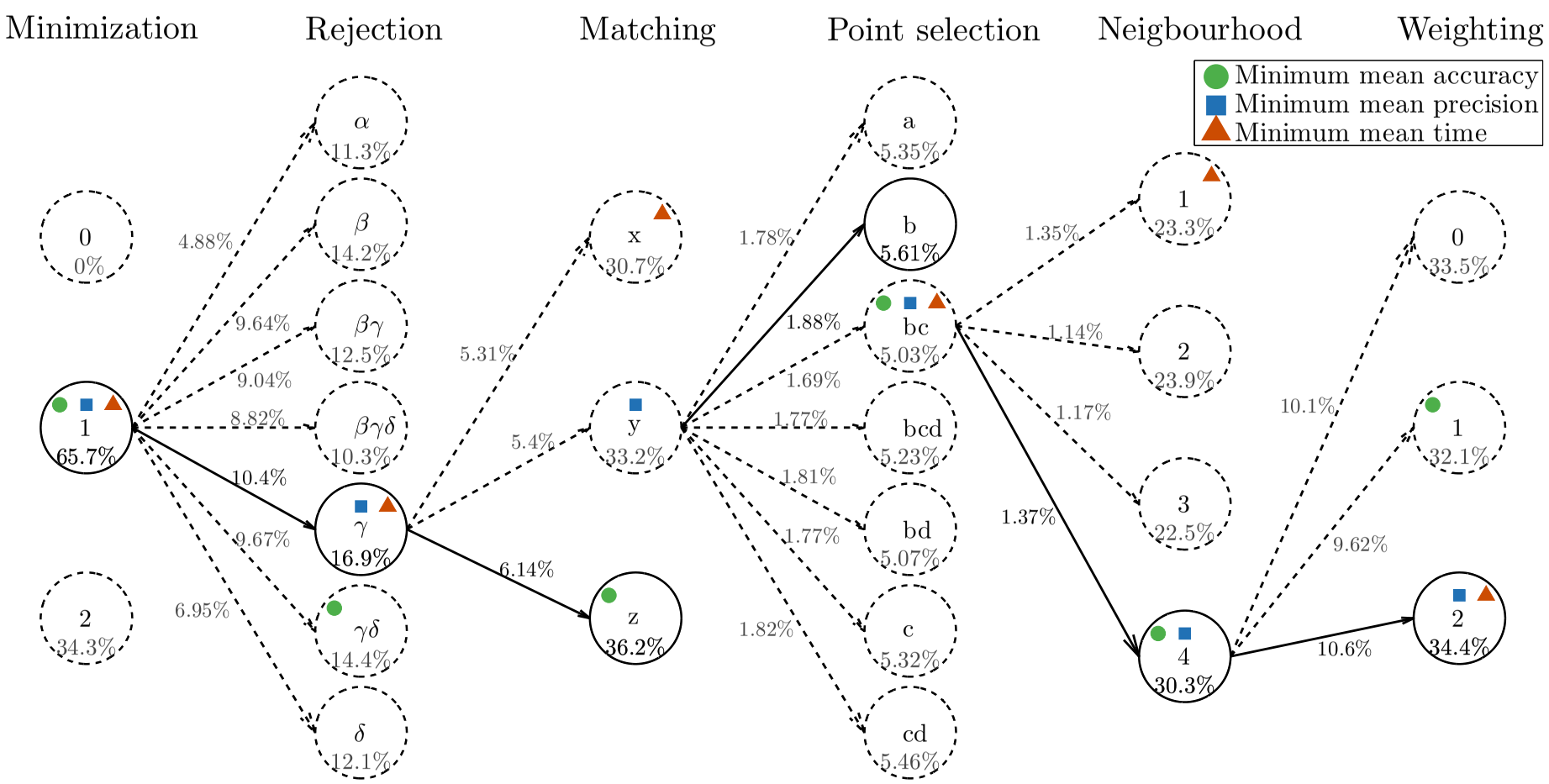
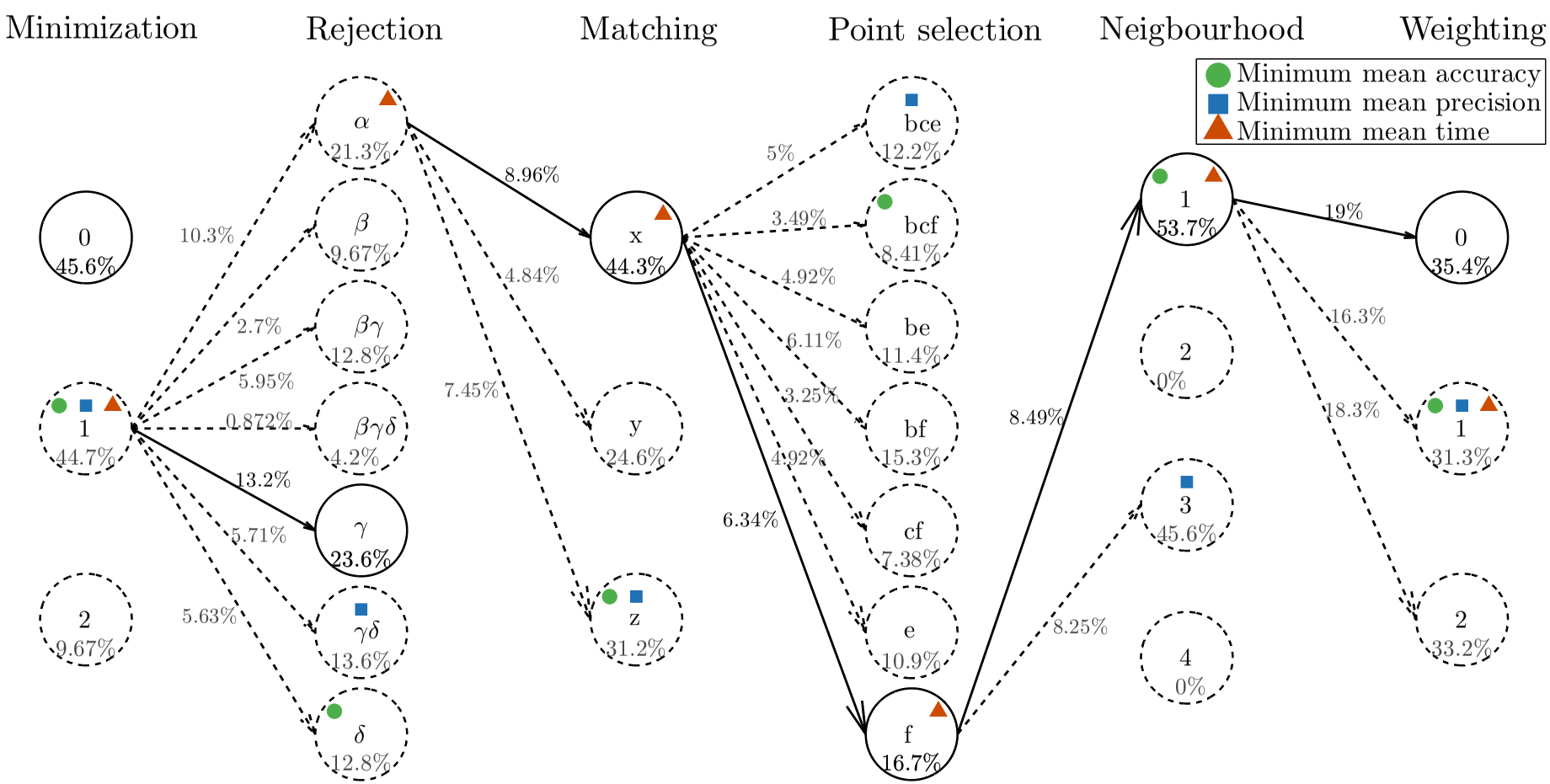


Figure 4.7: Method relationship diagram within the precision tolerance.



The 6.08% of 20,736 ICP variants are within the computational efficiency tolerance for the three scenes ($rct \leq 3$).

Figure 4.8: Method relationship diagram within the relative time tolerance.

Observation 4: *There is no single best ICP variant as measured by accuracy, precision, and computation efficiency.*

Nevertheless some general trends do show through. All of the accurate methods and 65.7 % of the precise methods use point-to-plane ICP for minimization. Point-to-plane minimization is furthermore used by 44.7 % of the computational efficient variants. Correspondingly, point-to-point minimization performs worst and is not recommended. This emphasis is further demonstrated by Fig. 4.9 which shows the average accuracy verses average precision for each variant. This conclusion is consistent with Pulli (1999) and Rusinkiewicz and Levoy (2001) both of whom found point-to-plane minimization to be more accurate than other methods. Figure 4.10 suggests that computational relative time is not correlated to the minimization method used.

Figure 4.5 shows how variants perform in accuracy per scene by showing the number of variants that fall within various accuracy bounds for scenes and minimization methods. Again the general conclusion is that point-to-plane is preferred, it being implicit in this conclusion that precision is correlated to accuracy. Interestingly, Fig. 4.5c shows that generalized minimization performs well for Scenes A and B but poorly for Scene C.

Observation 5: *Minimization is best performed by point-to-plane ICP and is preferred over generalized ICP and point-to-point ICP.*

Observation 6: *There is a preference to match points using the nearest neighbour enhanced by normals based on this method being the most prevalent and the most accurate ICP variants.*

From Fig. 4.6, 66.7 % of the most accurate variants use nearest neighbour enhanced by normals (Feldmar et al., 1995) for point matching. This appears to be the most accurate variant on average. But while precision is generally correlated with accuracy, the most prevalent point matching method (36.2 %) in the group of precise ICP variants is nearest neighbour enhanced by normals, although a significant proportion (30.5 %) are nearest neighbour enhanced by moment invariants (Sharp et al., 2002). The most common point matching method (44.3 %) among the fastest variant is, not surprisingly, nearest neighbour matching (Besl and McKay, 1992), this being the simplest algorithm.

The variation of matching methods across all ICP variants is shown in Fig. 4.11. It shows that nearest neighbour enhanced by normals has the best overall accuracy performance. Figure 4.12 shows that rct does not generally discriminate on the preferred point matching method.

Whilst point matching is an important step in ICP, none of the three algorithms considered is remarkably better than the others The best overall performance is achieved by the nearest neighbour with normals algorithm, based on being most preferred for its prevalent in the most accurate variants.

Observation 7: *Pair rejection improves scan matching. Rejection by angular deviation can be used effectively in combination with rejection by adaptive distance and rejection by worst distance percentage.*

None of the most accurate variants involve no rejection methods (Method α). This suggests that rejection is an important step in the registration process. Among the set of most accurate methods 100 % employ the rejection by angular deviation algorithm (Method γ) (Rusinkiewicz and Levoy, 2001), in combination with rejection by adaptive distance (method δ) (33.3 %) (Zhang, 1994) or rejection by worst distance percentage (Method β) (11.1 %) (Rusinkiewicz and Levoy, 2001). The three methods (Methods $\beta\gamma\delta$) applied together account for 55.6 % of the most accurate variants. Method γ also appears in 53.8 % of the most precise variants in combinations with Methods β and δ.

Four representative rejection methods are selected to plot in the Fig. 4.13 and Fig. 4.14. Figure 4.13a shows that the no rejection approach is far from the accuracy tolerance zone. Rejection by a percentage of the worst pair distances improve the variants in the point-to-point zone ($\nu > 2\,m$) and as expected, angular deviation rejection doesn't have any influence in the same zone. However, angular deviation rejection improves the performance considerably in the generalized minimization zone ($\nu > 1\,m$ and $\nu < 2\,m$). Figure 4.6 shows that the combination of angular deviation rejection and adaptive distance by variance rejection is the most prevalent rejection method in the accuracy tolerance zone and with the best accuracy performance.

Figure 4.14 shows that the computational time doesn't vary significantly among rejection methods. Figure 4.8 shows that no rejection is the fastest rejection method on average, however angular deviation rejection has slightly more variants in that zone than no rejection at all.

Observation 8: *Bounded radius is the preferred method for neighbourhood selection.*

From Fig. 4.6, all of the most accurate variants employ the density bounded radius (Lalonde et al., 2005) for neighbourhood selection making it a compelling choice among the variants considered. This algorithm is also the most prevalent (30.3 %) among the most precise variants. However, this method doesn't appear among the fastest methods (Fig. 4.8).

Figure 4.16 shows that the slowest method is neighbourhood adapted by point density followed by bounded radius. Neither of these methods have variants in the relative time tolerance region. As expected, a constant neighbourhood is the fastest method.

It can be concluded that the bounded radius methods outperforms other neighbourhood selection methods in terms of accuracy, however this method is intrinsically slow.

Observation 9: *There is no clear preference among point selection methods although outlier removal, dimensionality based selection and geometrically stable sampling appear commonly in the preferred*

variants.

Not withstanding that point selection is intensively studied and intuitively an important aspect of ICP, no strong preference emerges among the alternatives and combinations of alternatives. Figure 4.6 shows that there are only six method/combinations in the most accurate variants. The method with the worst performance is entropy feature filter with no presence in the most accurate variants. The most prevalent method in meeting average performance accuracy is the dimensionality based selection method (Gressin et al., 2012), either alone (Method f, 16.7 %) or in combination with outlier removal (Xie et al., 2004) (Method bf, 16.7 %) and the density filter method (Lalonde et al., 2006) (Methood cf, 22.2 %).

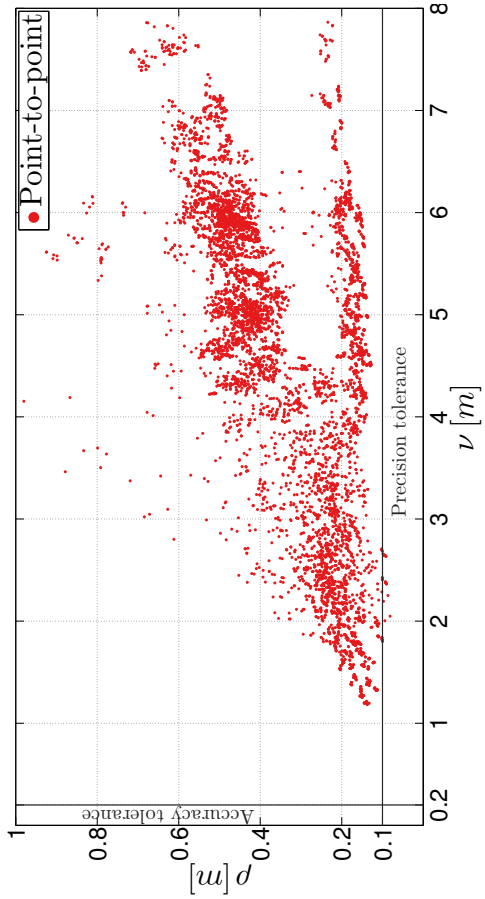
Four representative point selection methods are selected to plot in Fig. 4.17 and Fig. 4.18. Figure 4.17 shows that the pattern for the no point selection method, geometrical stable sampling, and outlier removal is quite similar. An exception is the entropy feature method which clearly presents a different pattern. Entropy feature filter notably improves the accuracy for $\nu < 1\,m$. This is the zone of generalized and point-to-point minimization. However it has less presence in the accuracy tolerance region. Figure 4.6 shows that the method with more presence in the accuracy tolerance region is the dimensionality selector, however the outlier removal with dimensionality selection obtains the best performance in this tolerance zone.

Figure 4.7 shows that outlier removal with a density filter has the best precision performance within the precision tolerance region. However, the method with most presence in this region is outlier removal.

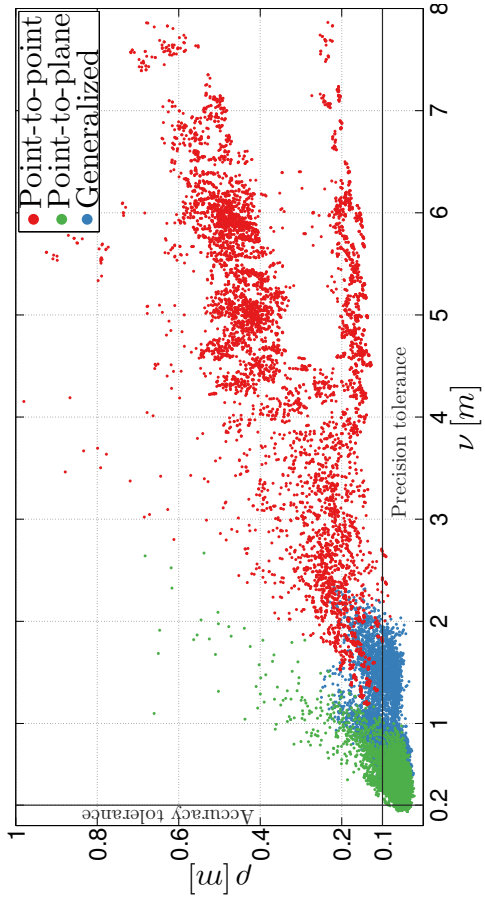
Observation 10: *There is no clear preference among weighting methods and this stage can be reasonably removed from the computation process.*

Weighting does not appear to significantly influence the performance of ICP variants. Among the most precise and accurate variants there is a slight preference for weighting by normal comparability. From Figs. 4.6 and 4.7 the relative proportion of methods distributed among the three weighting schemes are approximately equal for both most accurate and precise ICP variants. There is no clear reason for including this stage in the computation process.

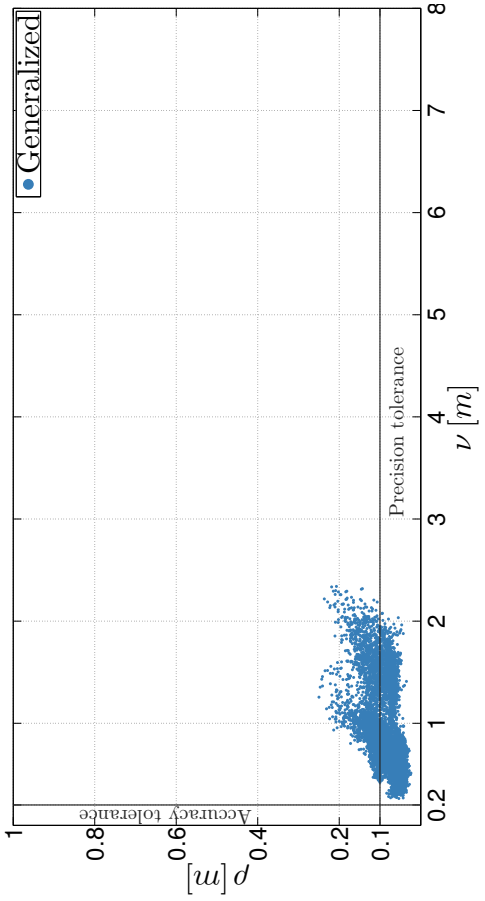
Figure 4.19 shows a similar pattern for the three weighting methods. Figure 4.6 confirms this tendency showing that the presence of each method is almost uniform. This same behaviour is observed in the precision tolerance region, Fig. 4.7, where each method has almost the same presence.



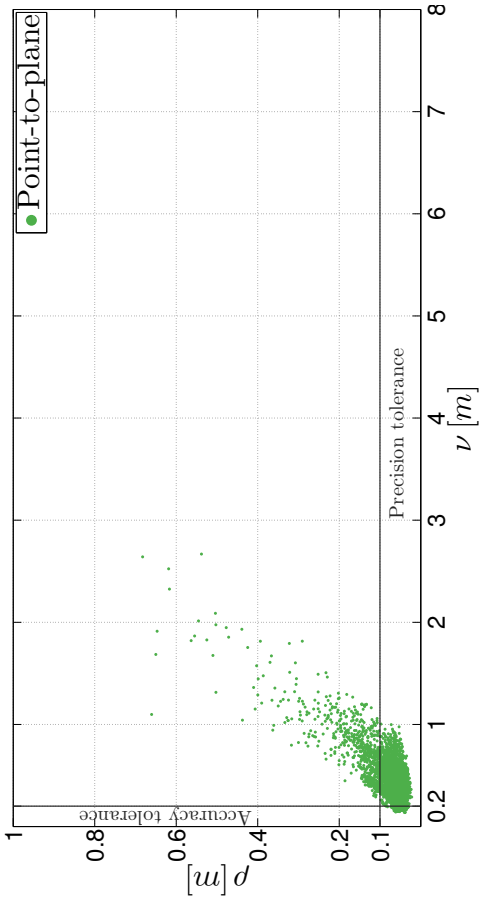
(b) Precision, ρ, vs accuracy, ν, for ICP point-to-point.



(a) Precision, ρ, vs accuracy, ν, for the three minimization methods.

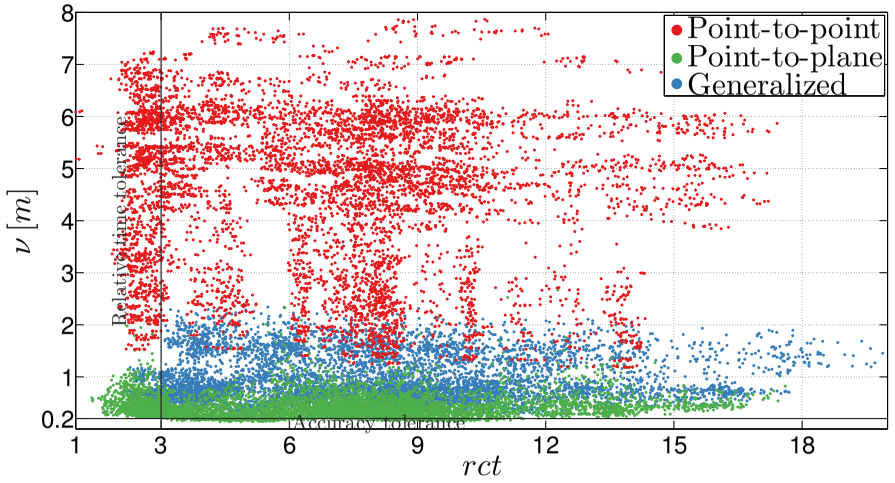


(d) Precision, ρ, vs accuracy, ν, for Generalized ICP .

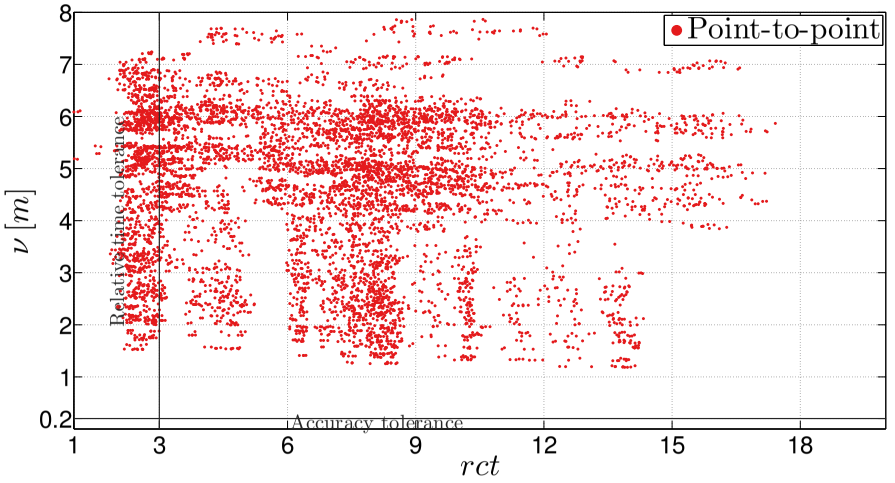


(c) Precision, ρ, vs accuracy, ν, for ICP point-to-plane.

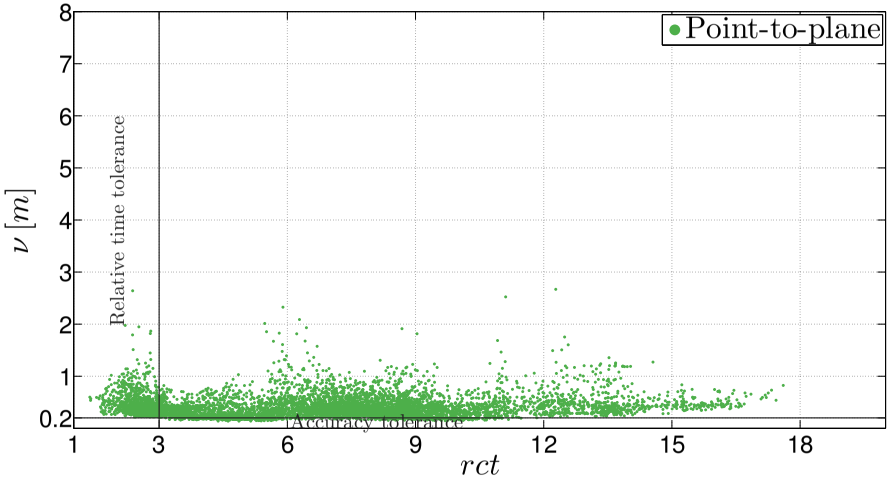
Figure 4.9: Performance of average precision-accuracy of 20,736 ICP variants, coloured by minimization method.



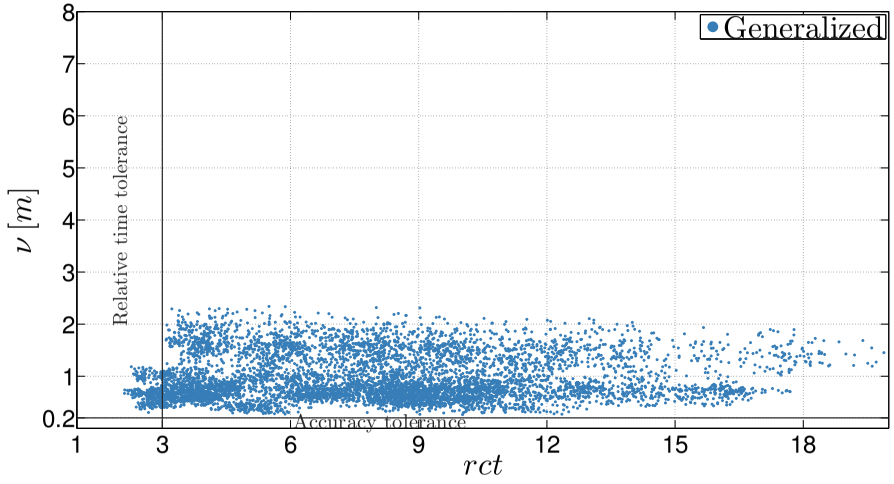
(a) Accuracy, ν, vs computational efficiency, rct, for the three minimization methods.



(b) Accuracy, ν, vs computational efficiency, rct, for ICP point-to-point.

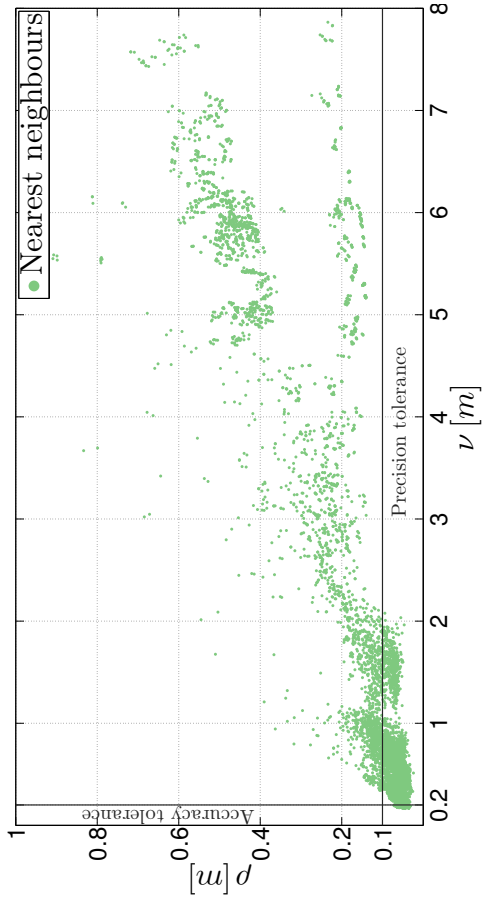


(c) Accuracy, ν, vs computational efficiency, rct, for ICP point-to-plane.

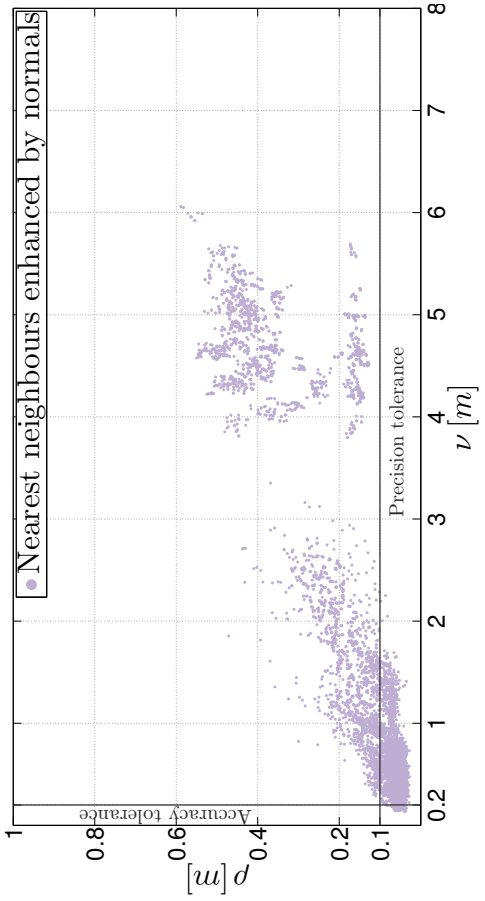


(d) Accuracy, ν, vs computational efficiency, rct, for Generalized ICP .

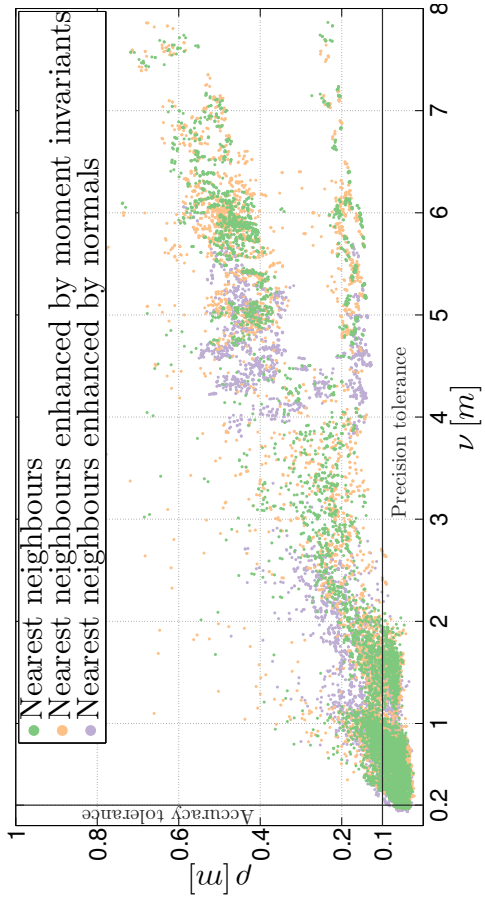
Figure 4.10: Performance of average accuracy-computational time of 20,736 ICP variants, coloured by minimization method.



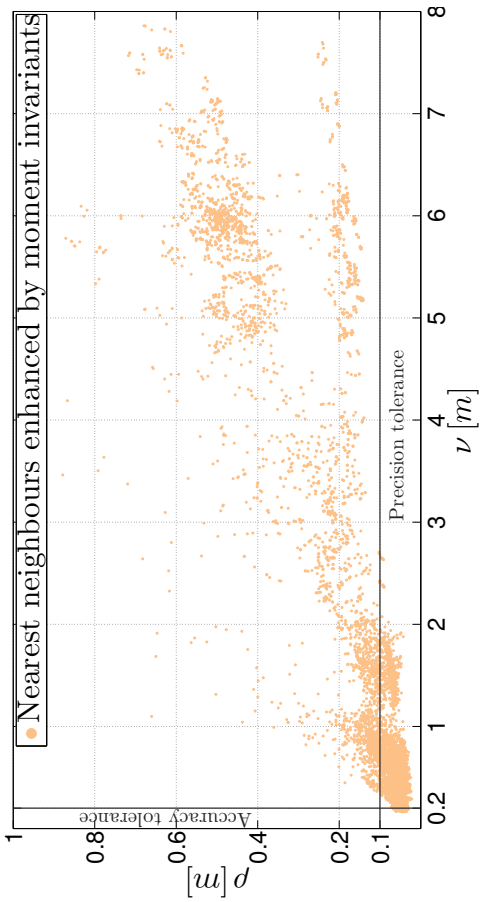
(a) All matching methods.



(b) Nearest neighbours.

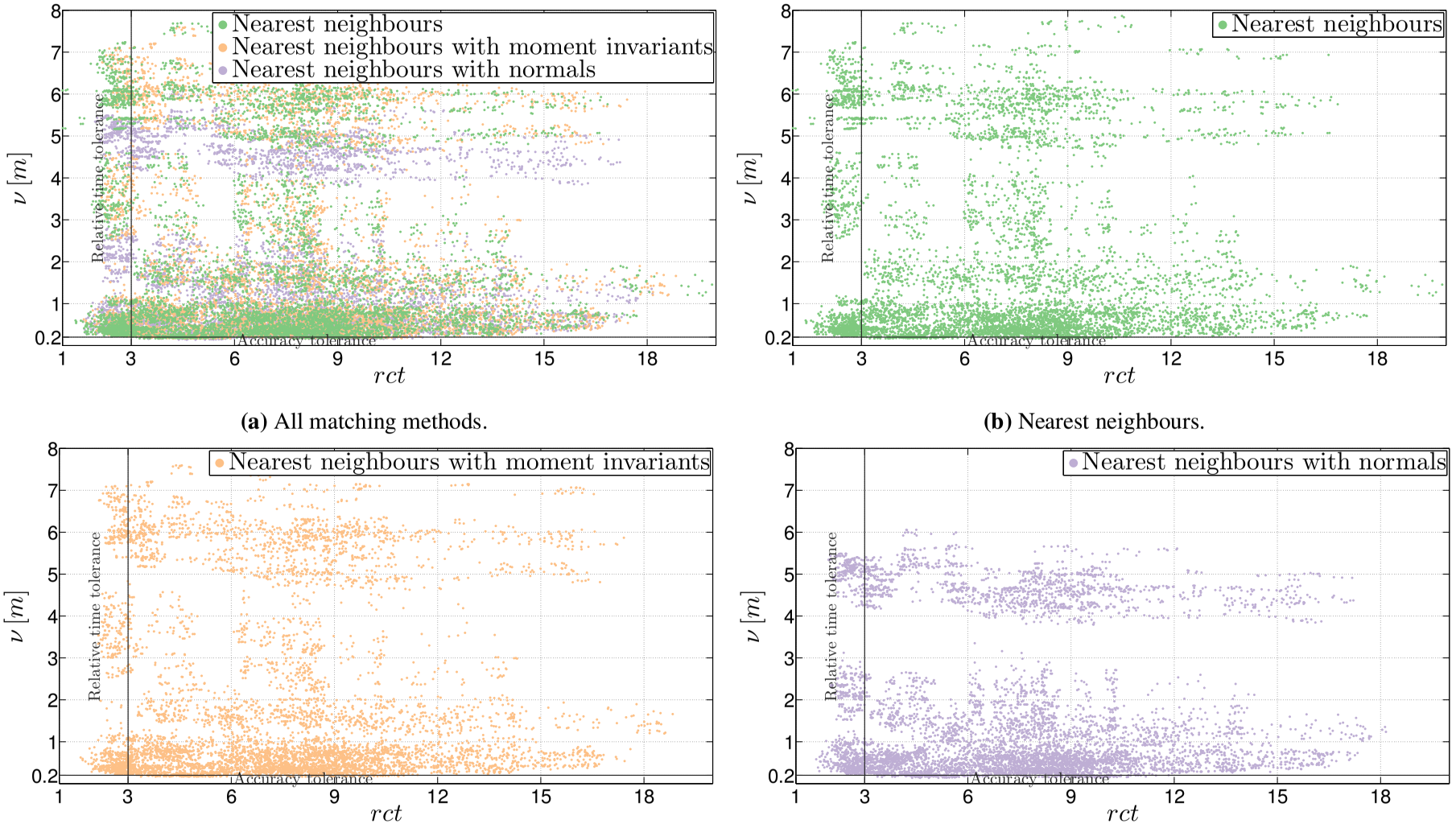


(c) Nearest neighbours enhanced by moment invariants.



(d) Nearest neighbours enhanced by normals.

Figure 4.11: Relation of precision, ρ, vs accuracy, ν, for point matching methods.



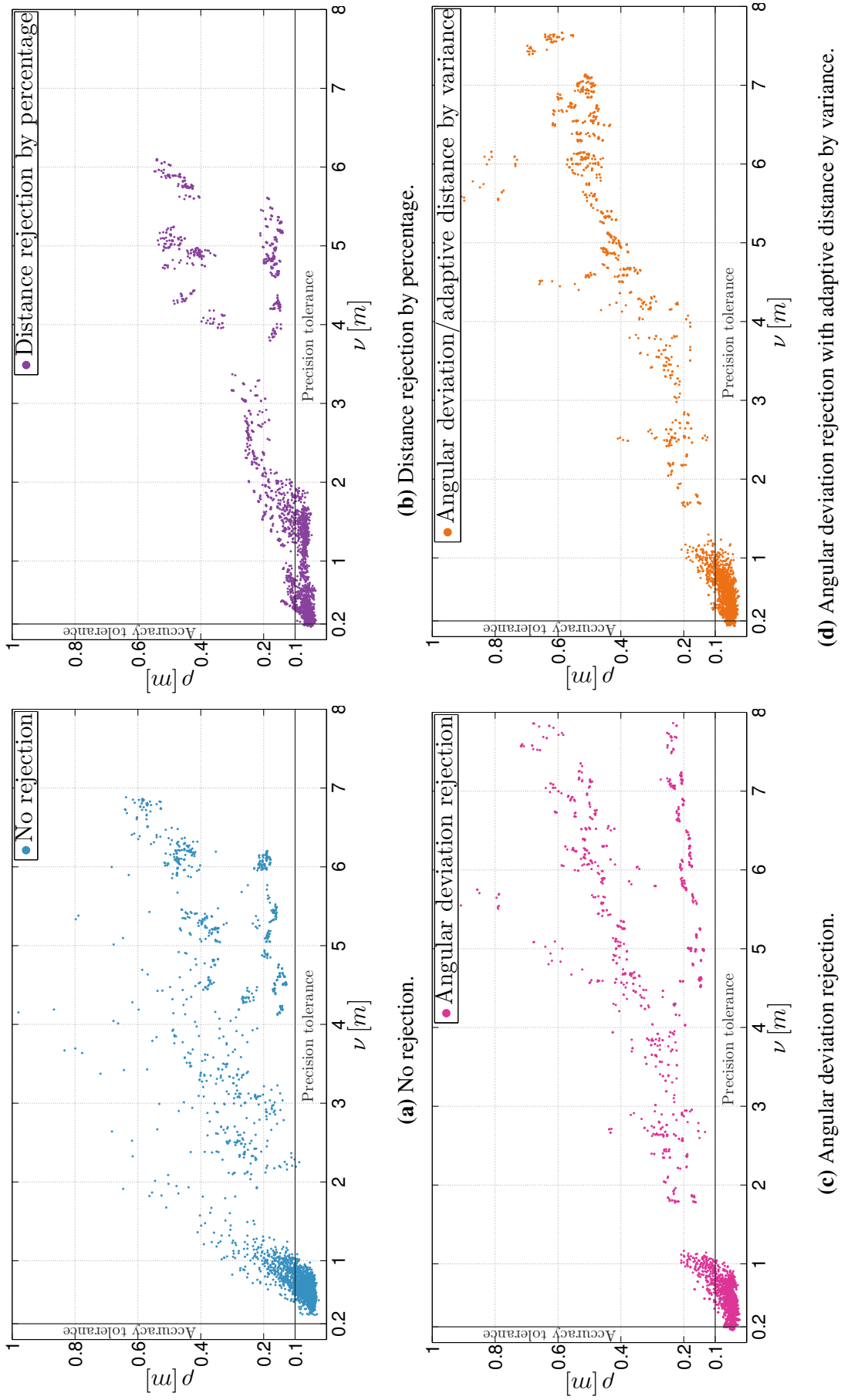
(a) All matching methods.

(b) Nearest neighbours.

(c) Nearest neighbours enhanced by moment invariants.

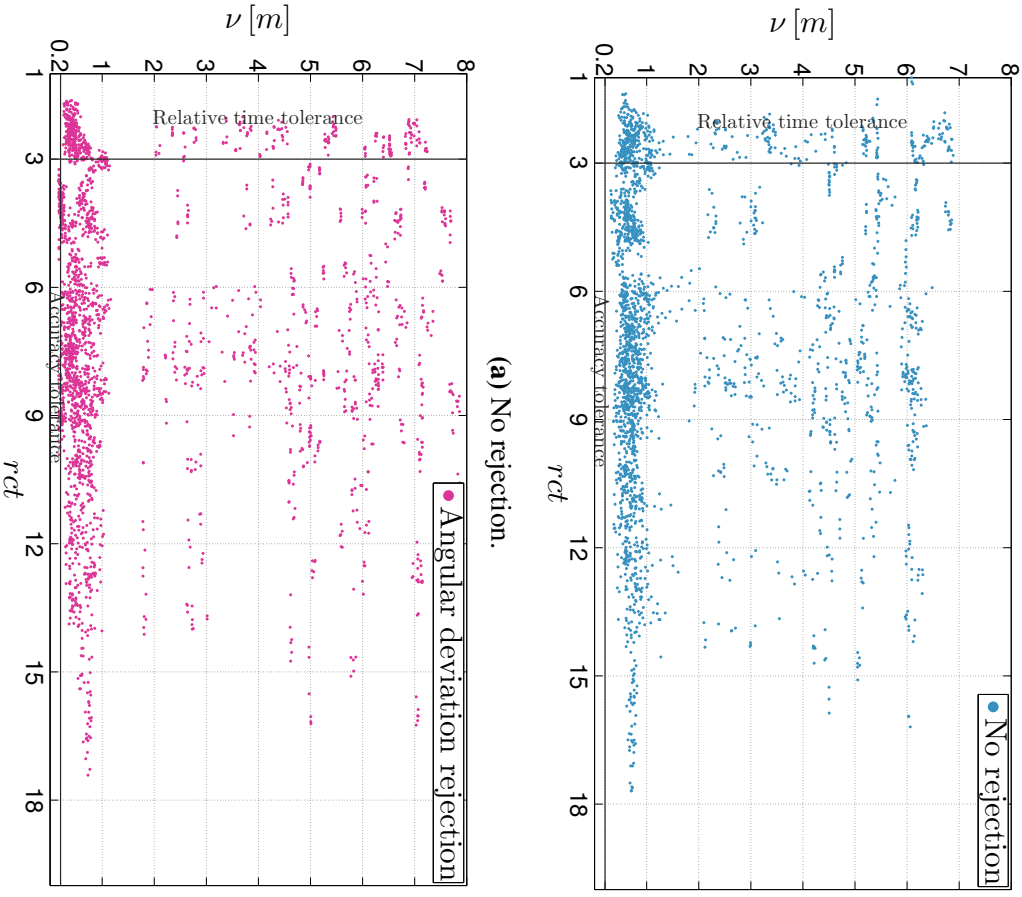
(d) Nearest neighbours enhanced by normals.

Figure 4.12: Relation of accuracy, ν, vs computational efficiency, rct, for matching methods.



(a) No rejection. **(b)** Distance rejection by percentage. **(c)** Angular deviation rejection. **(d)** Angular deviation rejection with adaptive distance by variance.

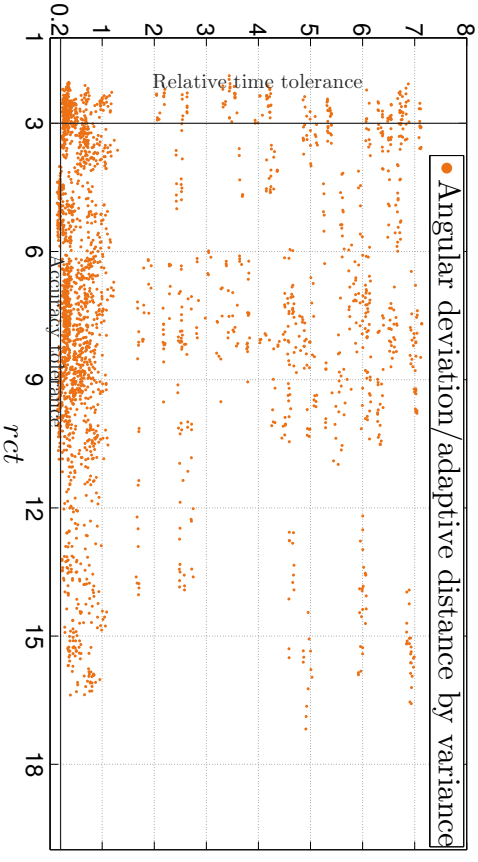
Figure 4.13: Relation of precision, ρ, vs accuracy, ν, for rejection methods.



(a) No rejection.

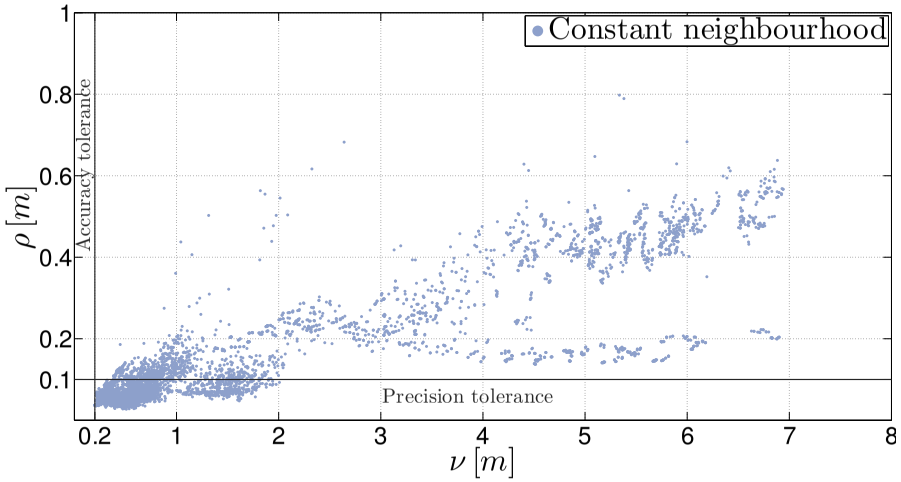
(b) Distance rejection by percentage.

(c) Angular deviation rejection.

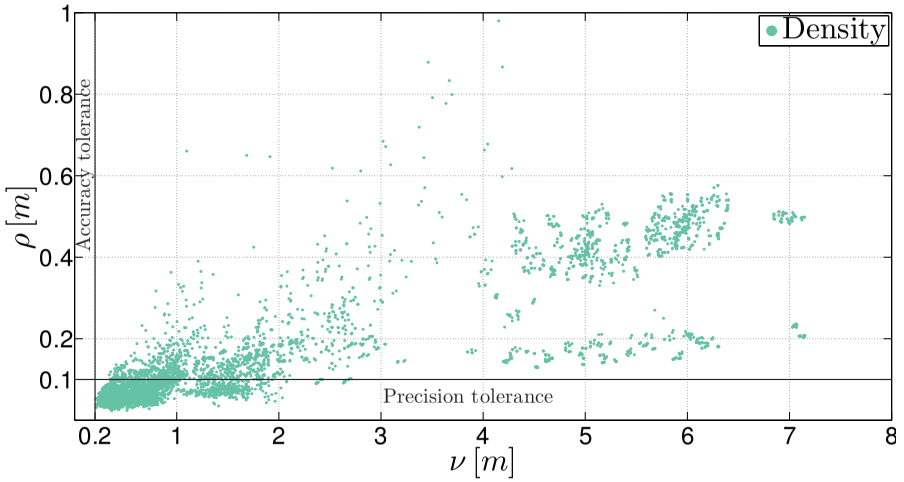


(d) Angular deviation rejection with adaptive distance by variance.

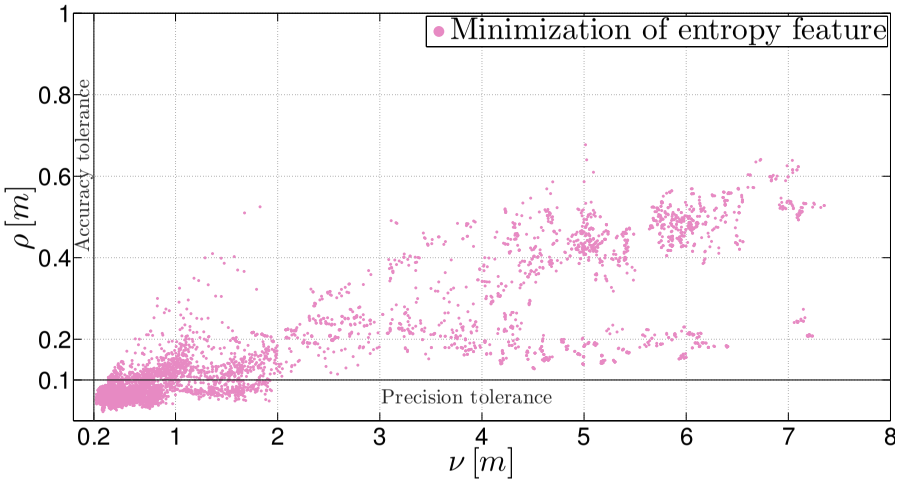
Figure 4.14: Relation of accuracy, ν, vs computational efficiency, rct, for rejection methods.



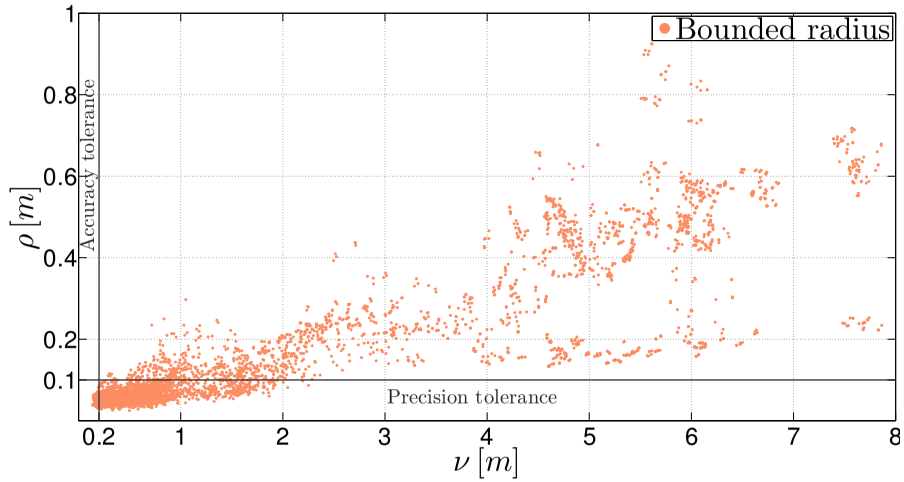
(a) Constant neighbourhood.



(b) Density.

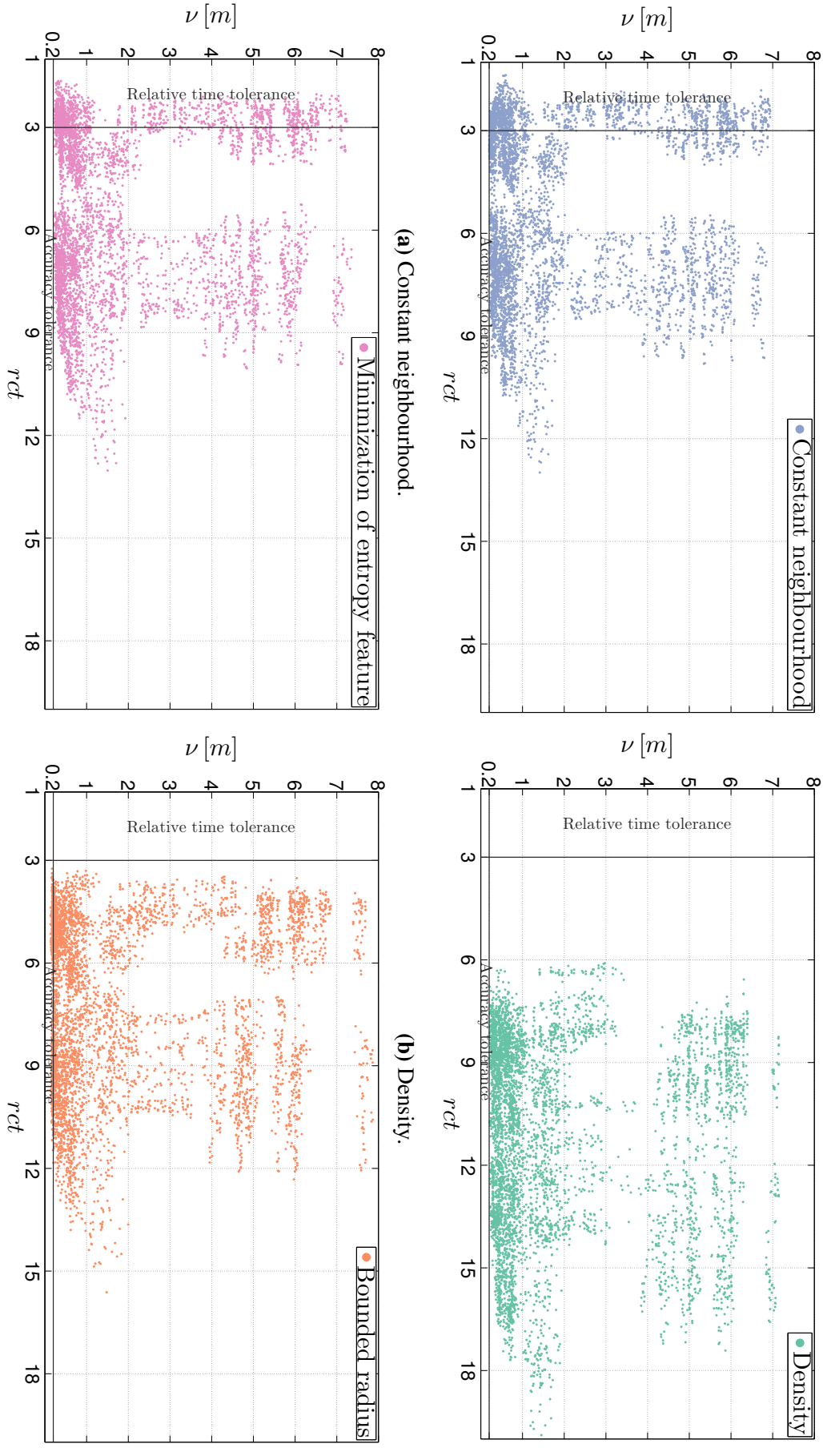


(c) Minimization of entropy feature.



(d) Bounded radius.

Figure 4.15: Relation of precision, ρ, vs accuracy, ν, for neighbourhood selection methods.



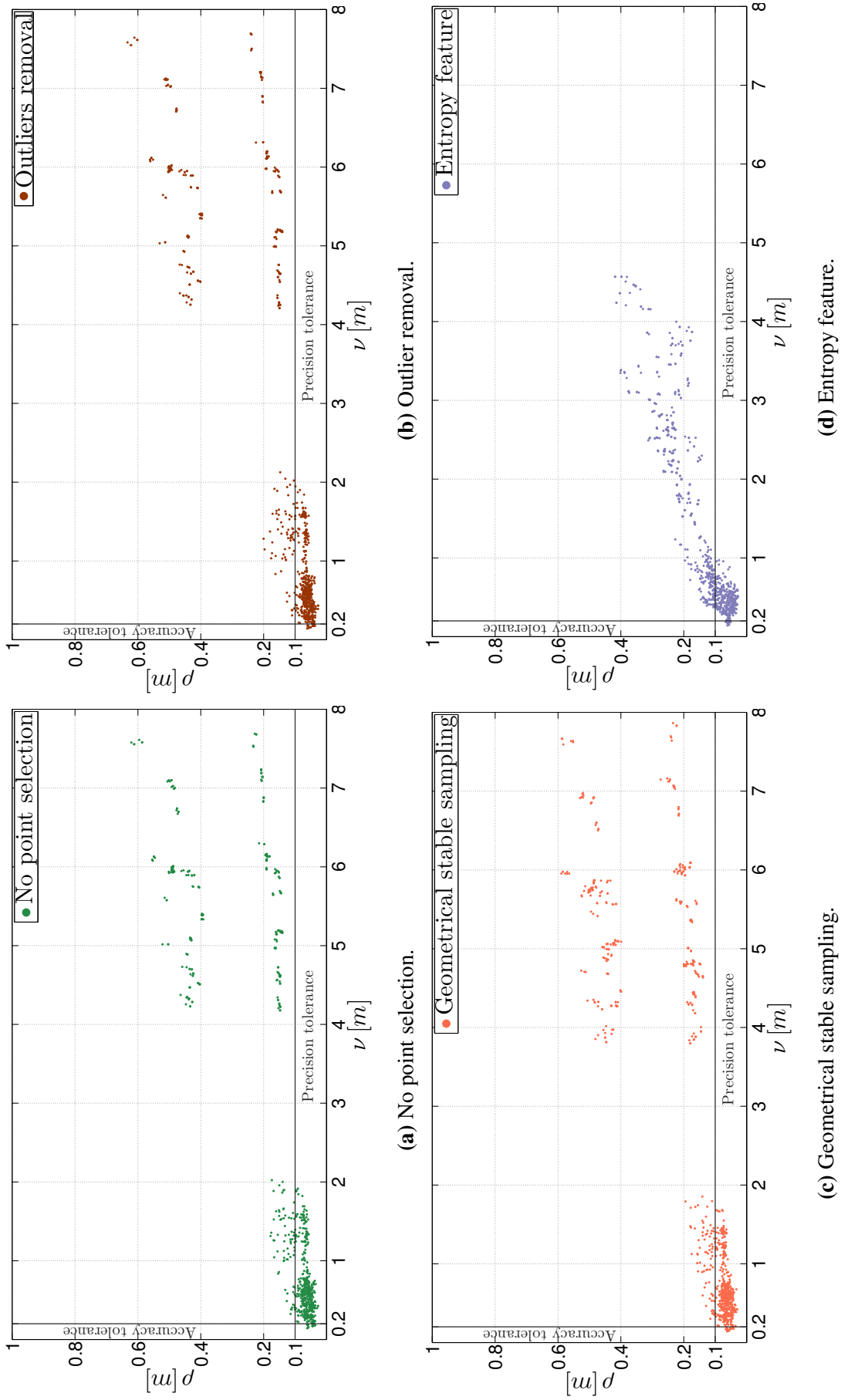
(a) Constant neighbourhood.

(b) Density.

(c) Minimization of entropy feature.

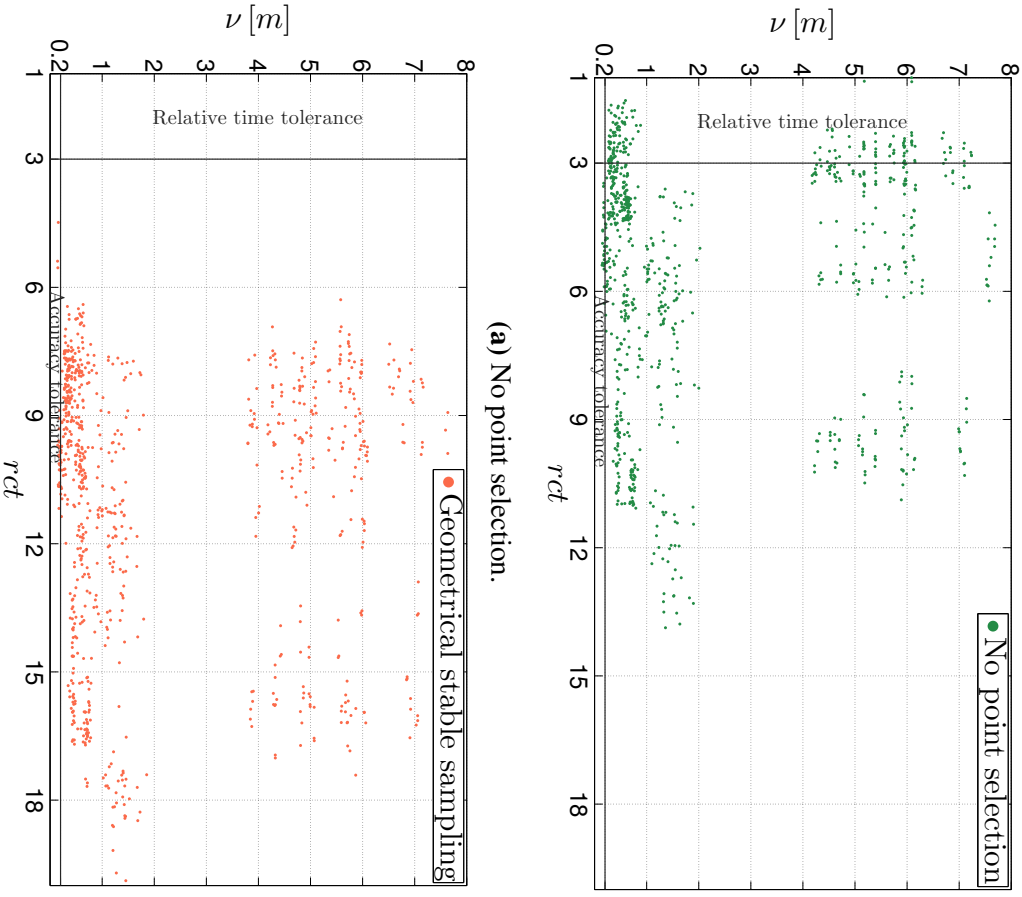
(d) Bounded radius.

Figure 4.16: Relation of accuracy, ν, vs computational efficiency, rct, for neighbourhood selection methods.

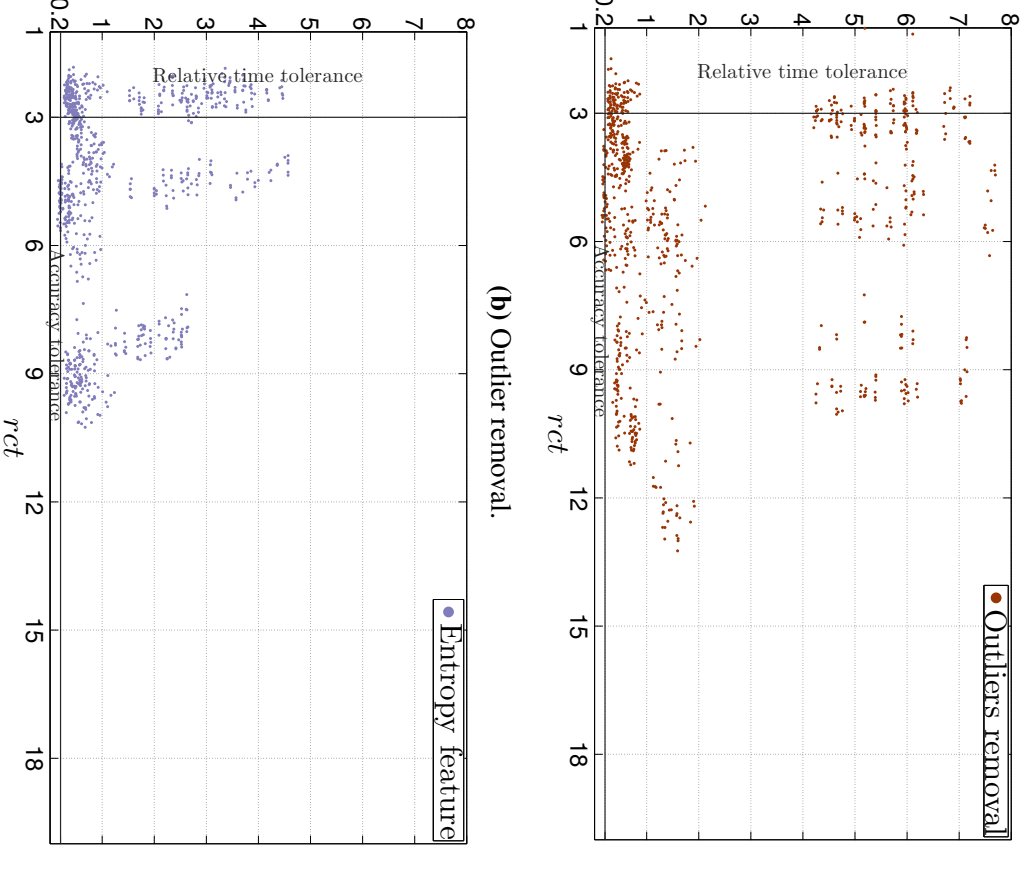


(a) No point selection. **(b)** Outlier removal. **(c)** Geometrical stable sampling. **(d)** Entropy feature.

Figure 4.17: Relation of precision, ρ, vs accuracy, ν, for selection methods.



(a) No point selection.

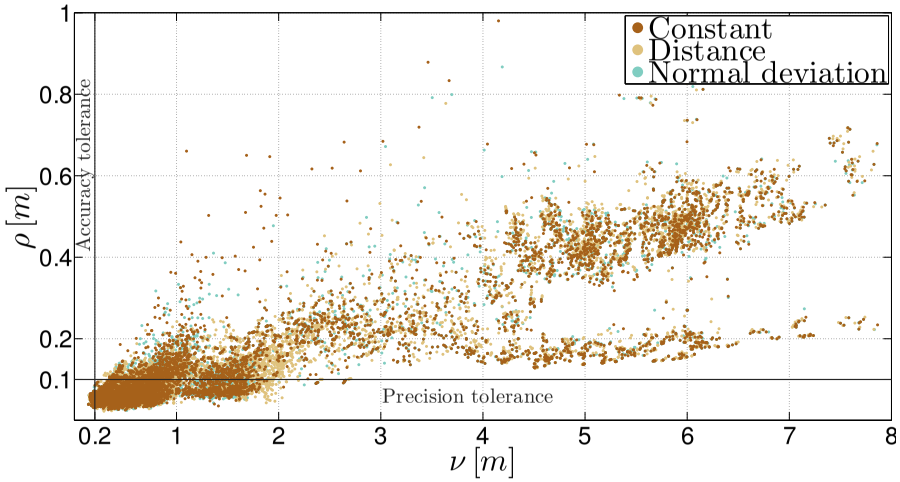


(b) Outlier removal.

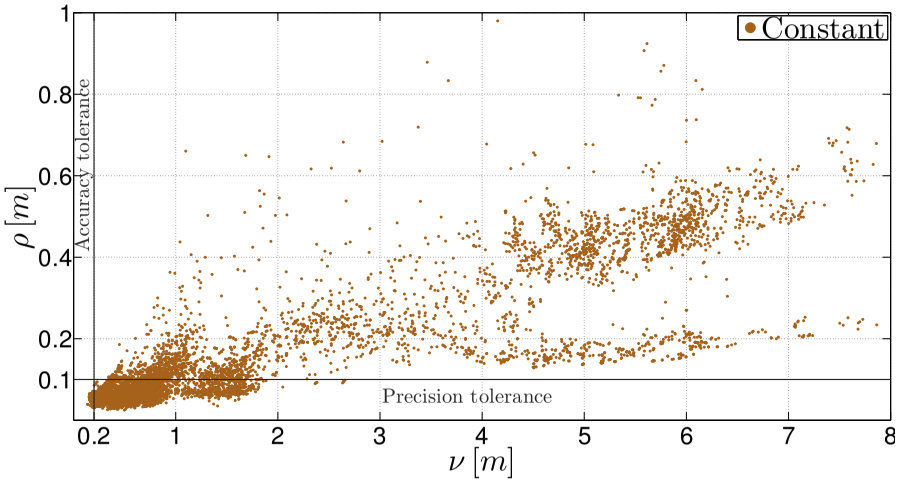
(c) Geometrical stable sampling.

(d) Entropy feature.

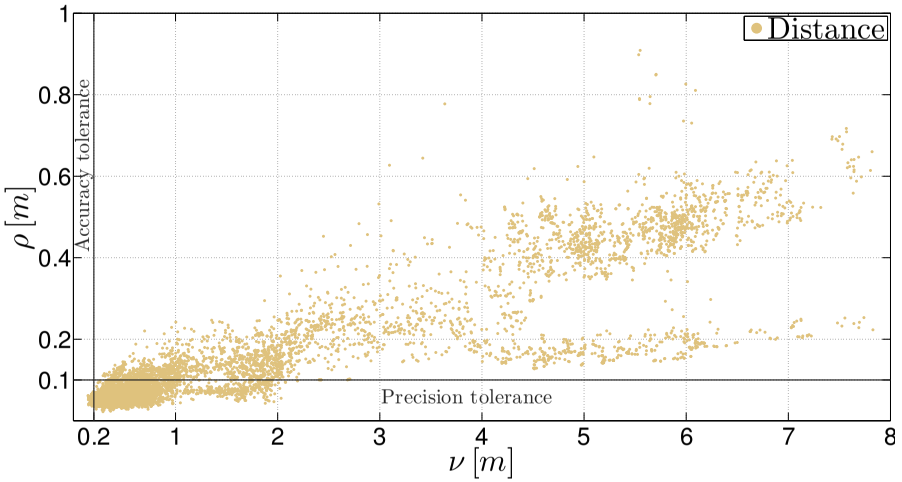
Figure 4.18: Relation of accuracy, ν, vs computational efficiency, rct, for neighbourhood selection methods.



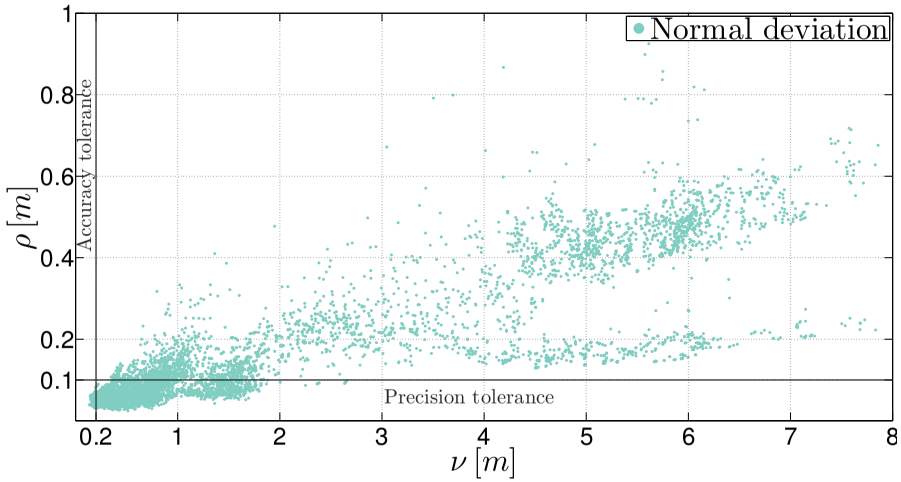
(a) All weighting methods.



(b) Constant weight.

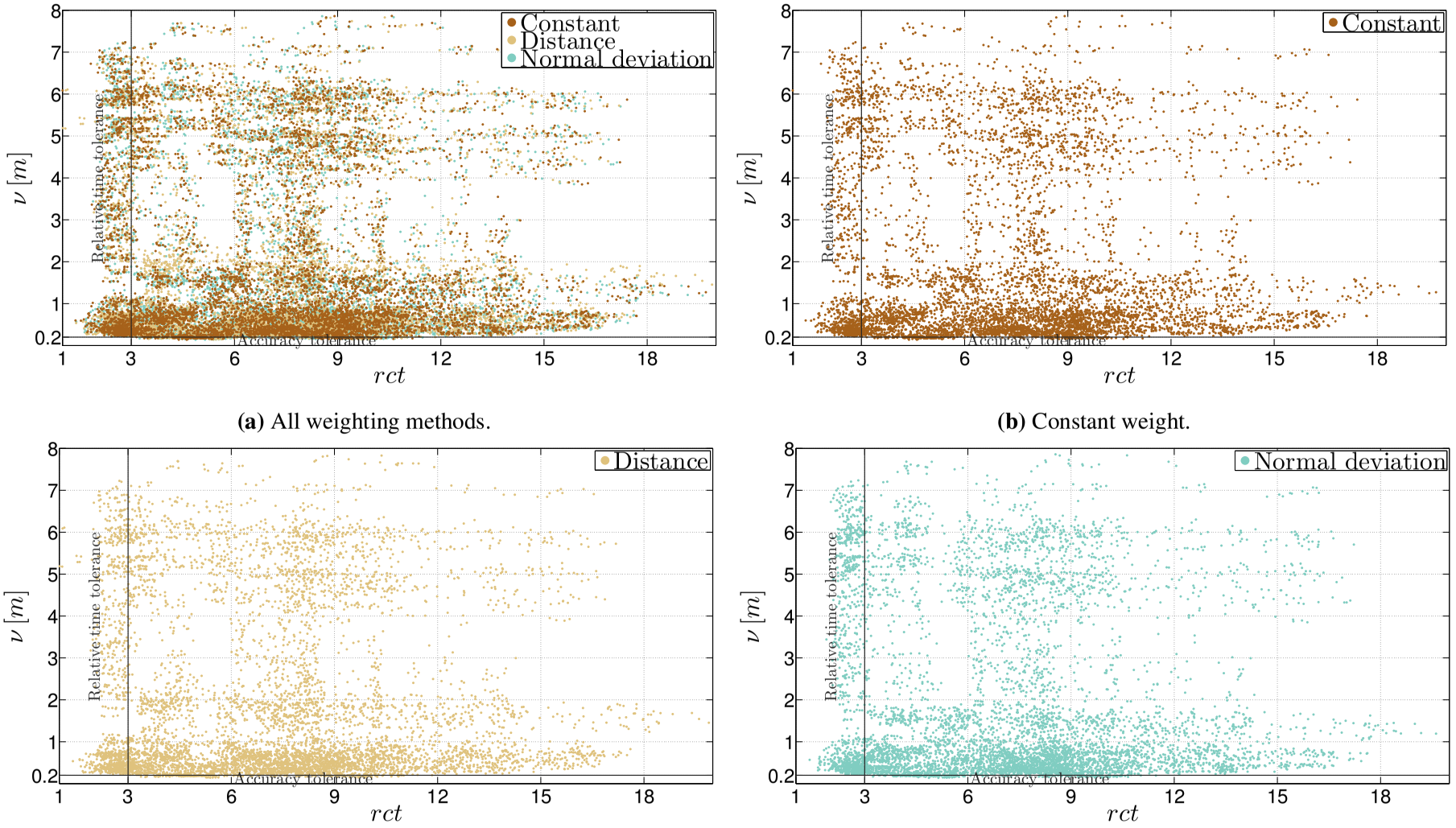


(c) By distance.



(d) By normal deviation.

Figure 4.19: Relation of precision, ρ, vs accuracy, ν, for weighting methods.



(a) All weighting methods.

(b) Constant weight.

(c) By distance.

(d) By normal deviation.

Figure 4.20: Relation of accuracy, ν, vs computational efficiency, rct, for weighting methods.

4.4 Conclusions

This chapter has consolidated research over the past twenty years in scan matching to establish a comparison of various ICP algorithms. The main finding of this chapter is that the way in which points are distributed in a point cloud has a significant influence on performance of the candidate ICP variants used for scan matching. It is evident that there is no single preferred variant, however some general patterns do emerge.

Specifically:

- Minimization by point-to-plane distance of Chen and Medioni (1992) outperforms other distance minimization methods.
- Point matching is best achieved using the nearest neighbour with normals algorithm of Feldmar et al. (1995).
- Rejection is an important stage of the ICP computation and appears to be most effectively achieved by the angular deviation method of Rusinkiewicz and Levoy (2001) alone or in combination with the adaptive distance method of Zhang (1994) or the worst percentage distance rejection method of Rusinkiewicz and Levoy (2001).
- Neighbourhood selection should be completed using the bounded radius method of Lalonde et al. (2005).
- There is no clear preference for point selection methods.
- There appears to be no clear benefit in point weighting.

These findings are consistent with the observations of Rusinkiewicz and Levoy (2001) and focus attention on the need to find means for enabling robust scan matching across different point cloud distributions. It is not clear how this might best be done but given no single method is best across all point clouds this argues for the need to find adaptive methods that can assemble a good performing variant given a point cloud or new methods that can tune themselves to the circumstances of a point cloud. To this latter point, the entropy based ideas presented in Gressin et al. (2012) have a strong appeal in so far as they look to measure the way in which the point cloud is distributed. This looks to be an avenue worthy of further exploration.

The time taken to complete an ICP scan match is clearly a significant issue. None of the methods considered meet the accuracy, precision, and computational efficiency thresholds. This is because the dynamic point neighbourhood methods produce the most accurate and precise matches but are also the most computationally expensive stages of the matching process.

The overall objective in this chapter has been to provide guidance to the implementers of ICP algorithms for scan matching. This investigation has focussed on a specific environment (the terrain of open-pit mining) for map building applications. However, much of this is common in other settings that have irregular structure, such as natural terrain, and these findings extend *mutatis mutandis* beyond the confines of this investigation.

Chapter 5

Three novel ICP variants

5.1 Introduction

Chapter 4 studied the performance of ICP variants against a reference data set considering accuracy, precision and computational efficiency and found, not unexpectedly, that some variants perform significantly better than others and that there is no single best variant that performed well for all metrics in all data. Useful guidelines emerged supporting the selection of preferred methods for minimization, matching, rejection and neighbourhood selection for the chosen terrain environments, and the limited benefit of existing point selection and weighting methods for terrain scan matching was identified.

Importantly, Chapter 4 identified that the most important factor determining the performance of the ICP algorithm for scan matching are the point clouds themselves. An ICP variant that performs effectively in matching a scan sequence in one scene can perform poorly against another scene. This observation motivates the work of the current chapter which has as its aim to develop methods that support more accurate, precise, and robust scan matching over different point clouds.

Several of the methods explored in Chapter 4 employed so-called point features to improve correspondences between the points of two clouds. Surface normals are an example of point features. More generally, in scan matching, point features can be thought of as information associated with each point of a cloud describing the local properties. This information is used to aid the establishment of correspondence between clouds.

This chapter focusses on the use of entropy-based methods for computing features. Two novel methods are proposed that act at different stages of the ICP process. The first is a point selection method based on what is termed *eigentropy*. The second method is a novel pair rejection strategy that uses the eigentropy of the reference point cloud to unilaterally reject pairs. The chapter also explores the performance of a simple (non-entropy based method) to select matching points based on normal angle deviation.

The methodology developed in Chapter 4 is extended to include these new methods to determine their capability to improve ICP performance in terms of average performance and robustness.

The structure of the chapter is as follows. Sections 5.2, 5.3 and 5.4 review background ideas and related work in using geometry features for registration. Sections 5.5 and 5.6 present the novel methods introduced in this chapter. Section 5.7 shows the evaluation methodology. Section 5.8 presents results and discussions. Section 5.9 summarises the conclusions.

5.2 Features in point cloud data

The starting point for this chapter is to look closer at methods used to identify features in point clouds.

A point in three-dimensional space can be described by three Cartesian coordinates: $\mathbf{p}_i = \{x_i,\ y_i, z_i\}$. A point cloud is a collection of individual points. The term "feature" is used to describe information that characterizes the cloud. Many different features have been proposed and these are reviewed, for example, in Jain and Dorai (2000); Tangelder and Veltkamp (2008); Qin et al. (2008).

Table 5.1 shows a categorization of different feature methods organized hierarchically by their sophistication.

Table 5.1: Point selection methods.

Category	**Feature**
Intrinsic to the sensor	Intensity Colour Temperature
Geometric primitives	Lines Polygons Planes
Geometric point	Curvature Normal Integral volume descriptor Moment invariants Eigenvalues Tensor voting Entropy feature
Rich descriptors	Zernike moments SURF descriptors Integral volume descriptor

Continued on next page

Table 5.1 – *Continued from previous page*

Category	**Feature**
	SIFT descriptors
	Heat Kernel Signature
Histogram based	Point Feature Histogram
	Fast Point Feature Histograms
	Signature of Histogram Orientations
	Spin images

The most basic features are so-called intrinsics features that are usually characteristics associated with the sensor. Many LiDAR-type range scanners provide an intensity measurement for each point that characterizes the strength of the associated return. Intensity is correlated to the reflectivity of the surface being ranged.

Intensity features can be used in various ways. Akca (2005) presents an algorithm for least squares matching of overlapping three-dimensional surfaces. The author incorporates the return intensity directly to the matching algorithm to improve robustness when there is a lack of sufficient geometric information in the point cloud. The method establishes what the authors term quasi-surfaces, $\mathbf{p}_i^q$, that combine point measurements with the intensity information. Normals, $\mathbf{n}_i$, are computed for each point, $\mathbf{p}_i$, and combined with intensity, c_f, to produce a quasi point, $\mathbf{p}_i^q = \mathbf{p}_i + c_f \lambda \mathbf{n}_i$, where λ is a scalar factor. Surfaces fitted to the quasi point clouds are matched using a generalized Gauss-Markov model, minimizing the sum of squared Euclidean distances between surfaces. The approach is claimed to improve robustness although it suffers in two respects: (i) range errors associated with high intensity returns tend to be larger than those low intensity returns (Phillips et al., 2014); and (ii) intensity measurements are subject to variation, depending for instance, on the angle of incidence of the LiDAR ray to the surface.

Hebel and Stilla (2007) use intensity information to constrain the correspondence estimation. They found that the difference of intensity between points is erratic due to the active illumination of the scene. Along similar lines, colour has been introduced into a variant of ICP by adding a weighted *hue value* as the 4^{th} dimension (Men et al., 2011). Colour has also been used for an extension of the normal distributions transform by Huhle et al. (2008) who show an improvement in the robustness of the algorithm.

Geometric primitive features (planes, lines, polygons) have an important role in scan matching in highly structured environments. This is particularly useful for indoor scenes where walls, floors, and

ceilings are generally flat surfaces that are often orthogonal. Cox (1991) makes use of this structure by a method that matches the input points against line segments extracted from the reference point cloud. Zhang and Faugeras (1991) propose a method to find the transformation that minimizes the error between two line segment sets. Yao et al. (2010) expand the registration by line segments to three-dimensional point clouds, using RANSAC (random sample consensus) (Fischler and Bolles, 1981). The extraction of robust geometrical primitives from an unstructured point cloud, such as those representing mining terrain, is a difficult task.

Rich feature descriptors include Zernike moments (Khotanzad and Hong, 1990), scale-invariant feature transform (SIFT) (Lowe, 2004) and speeded up robust features (SURF) (Bay et al., 2006). These methods typically have providence in methods for image registration and are characterized by providing richer information about the point neighbourhood. Houshiar et al. (2013) compare image features and descriptors for the registration of panoramic reflectance images. They show that SIFT and SURF algorithms present the best performance for automatic registration. Extensions of these methods for 3D point clouds are presented in Scovanner et al. (2007) and Flint et al. (2007) for SIFT and SURF, respectively.

Histogram methods include point feature histograms (PFH) and fast point feature histograms (FPFH) (Rusu et al., 2008), signature of histogram orientations (HOG) (Dalal and Triggs, 2005) and spin images (Johnson, 1997). Rusu et al. (2008) propose the PFH to enhance the discrimination of objects using the relationships between geometrical features. The method aims to find relationships between the query point and its neighbourhood by comparing their normal vectors. A Darboux frame is used to compare normals in a common frame, by obtaining four features that indicate the similitude of normals. These features are three angles (one for each axis) and the Euclidean distance between points. For k-neighbours, the method constructs $\frac{k(k-1)}{2}$ groups of these four features. The features are binned into a PFH histogram and the ratio of points in each bin is used to represent the shape of the neighbourhood. The selection of the geometrical shape is given by a training process in order to discriminate which bin configurations belong to a sphere, corner, plane, cylinder or edge class. For a detailed study of PFH and its extension FPFH, see (Rusu, 2010). For a complete review of the most popular histogram based features, see (Sukno et al., 2012; Behley et al., 2012). Note that these features have been successfully applied for coarse registration (Rusu et al., 2009; Brusco et al., 2005).

5.3 Computation of point cloud normals

The focus of this chapter is on geometric point features, that is, features associated with a given neighbourhood of each point in the point cloud. Normals and curvature are the basic geometric point features and they are obtained from geometrical relations within a neighbourhood of points.

Normals, generally, are calculated by finding the best fitted plane to a set of points. There are several

algorithms for this, see Badino et al. (2011) or Dey et al. (2005). Klasing et al. (2009) demonstrate that normals obtained by principal component analysis (PCA) are more robust (that is, less susceptible to noise) and can be computed faster than other methods. The quality of normals depends critically on the distribution of points in the neighbourhood. "Good normals" are critical for scan matching based on point-to-plane minimization.

For this work PCA is used to compute normals. It is worthwhile to briefly review the method of computation. The first step to calculate the normal vector of a point, $\mathbf{p}_i$, using the PCA procedure is to obtain the three-by-three covariance matrix, C_i, over a given neighbourhood of the point. The eigen-structure of C_i describes the distribution of points local to $\mathbf{p}_i$. The eigenvectors describe the principle axes about which neighbouring points are distributed. The variance of the point distribution over these axes are described by the eigenvalues. The normal is taken to be the eigenvector having the smallest eigenvalue corresponding to the direction with the least local point distribution. This direction is orthogonal to the other two more populated axes, *i.e.* these populated axes are the components of the best fitted tangent plane.

5.4 The eigen-structure of point clouds

More generally, the eigen-structure of C_i describes the distribution of points around $\mathbf{p}_i$ as an ellipsoid. The semi-major axis of this ellipsoid is the eigenvector associated with the largest eigenvalue and the square root of the largest eigenvalue is the semi-major axis length. The semi-minor axis is the eigenvector associated with the smallest eigenvalue, with the square root of the associated eigenvalue defining the semi-minor axis length.

If one of the three eigenvalues is large relative to the other two, the points are aligned to this axis and the resulting distribution is sometimes called a stick (Jolliffe, 2002). Where two of the three eigenvalues are much larger than the third, points are distributed on a plane (with the normal corresponding to the semi-minor axis). When all eigenvalues have similar magnitude, the points are distributed as a ball and said to be scattered. This relationship between eigenvalues and the geometry of point distribution is summarized in Table 5.2.

Table 5.2: Relations between the eigenvalues λ_i and the geometry of the point neighbourhood. Note: $\lambda_1 \geq \lambda_2 \geq \lambda_3$.

Eigenvalue relations	Geometric
$\lambda_1 \ggg \lambda_2, \lambda_3 \simeq 0$	Stick/line
$\lambda_1, \lambda_2 \ggg \lambda_3 \simeq 0$	Plane
$\lambda_1 \simeq \lambda_2 \simeq \lambda_3$	Ball/scatter

Classifying points directly from the magnitudes of λ_i is ambiguous because the relationship between

eigenvalues (Table 5.2) depends on numerical factors (Natale et al., 2010). A more manageable representation is provided in Medioni et al. (2000) via a second-order tensor representation for the covariance matrix C_i. The eigenvalue decomposition of C_i can be written as,

$$C_i = \begin{pmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \\ \mathbf{e}_3^T \end{pmatrix},$$

were $\mathbf{e}_i$ are the eigenvectors. In tensor notation,

$$\begin{aligned} C_i &= \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^T + \lambda_3 \mathbf{e}_3 \mathbf{e}_3^T \\ &= \underbrace{(\lambda_1 - \lambda_2)}_{\text{tensor}} \underbrace{\mathbf{e}_1 \mathbf{e}_1^T}_{\text{line}} + \underbrace{(\lambda_2 - \lambda_3)}_{\text{tensor}} \underbrace{\left(\mathbf{e}_1 \mathbf{e}_1^T + \mathbf{e}_2 \mathbf{e}_2^T\right)}_{\text{planar}} + \underbrace{\lambda_3}_{\text{tensor}} \underbrace{\left(\mathbf{e}_1 \mathbf{e}_1^T + \mathbf{e}_2 \mathbf{e}_2^T + \mathbf{e}_3 \mathbf{e}_3^T\right)}_{\text{scatter}} . \end{aligned}$$

West et al. (2004) presents a system architecture developed in a DARPA project for contextual cue extraction and target detection in three-dimensional point cloud data employing tensors. Whilst the work is not concerned with scan matching, it does look to establish the degree to which target detection, e.g. for air strikes, can be successfully addressed from the three-dimensional shape and geometry of point clouds through different tensor based features. The features identified include:

- Omnivariance = $(\lambda_1 \lambda_2 \lambda_3)^{1/3}$.
- Anisotropy = $(\lambda_1 - \lambda_3) / \lambda_1$.
- Planarity = $(\lambda_2 - \lambda_3) / \lambda_1$.
- Sphericity = $1 -$ Anisotropy
- Linearity = $(\lambda_1 - \lambda_2) / \lambda_1$.
- Eigenentropy = $-\lambda_1 \log(\lambda_1) - \lambda_2 \log(\lambda_2) - \lambda_3 \log(\lambda_3)$.

Later in this chapter it will be explored feature filters based on quantities similar to the eigenentropy of West et al. (2004).

Demantké et al. (2011) proposes what they call an entropy feature based on the tensor terms, as detailed in Algorithm 3.

Algorithm 3 Entropy features.

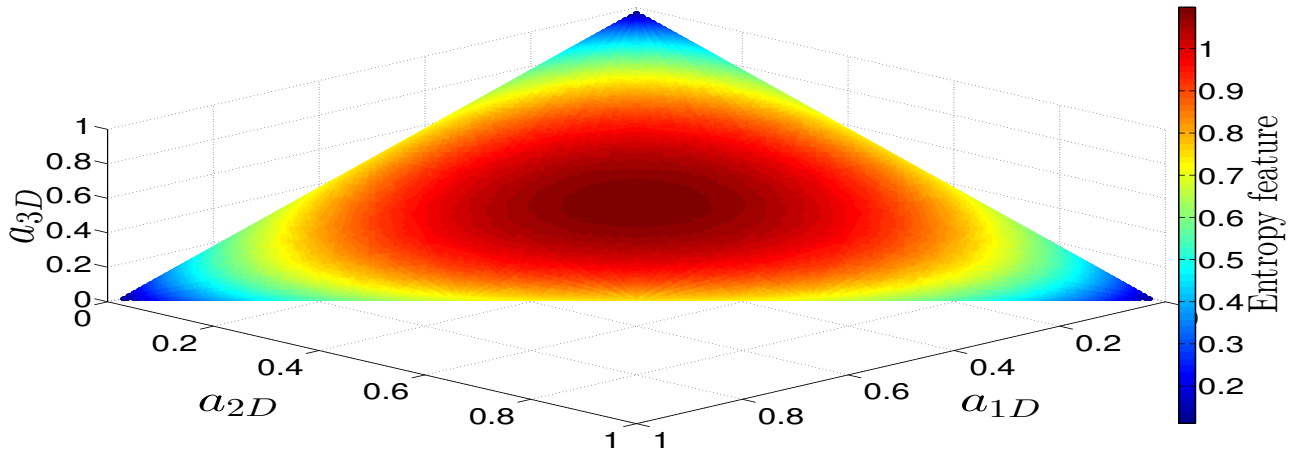
Require: Eigenvalues, λ_i, obtained by PCA computed from a given neighbourhood. σ_i defines the standard deviation of the eigenvector, $\hat{\mathbf{e}}_i$. a_{1D} represents the linearity dimensionality, a_{2D} represents the planarity dimensionality and a_{3D} is the scatter dimensionality.

Ensure: E_f defines the entropy feature.

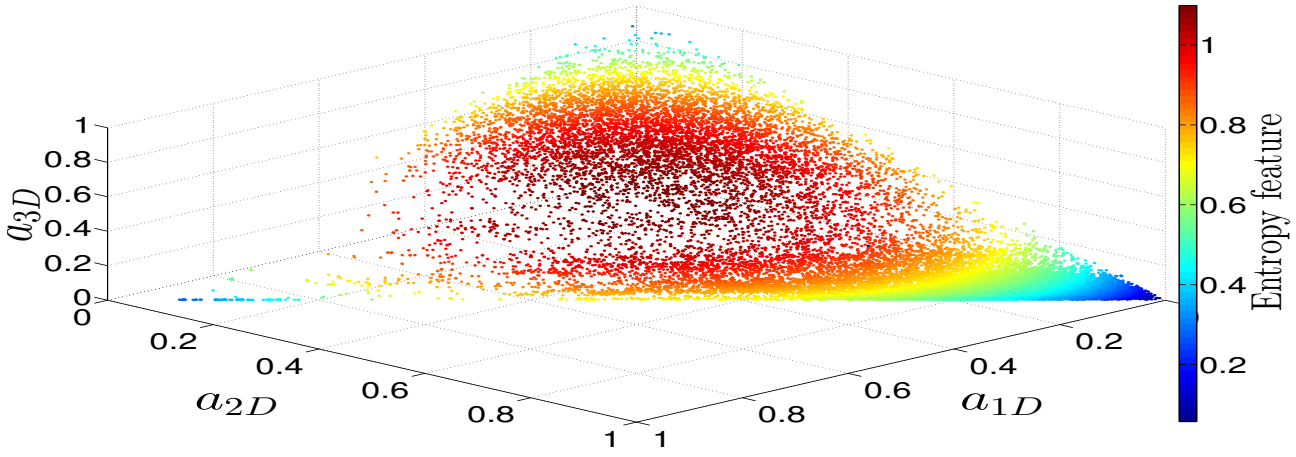
1: $\sigma_i = \sqrt{\lambda_i}$

2: $a_{1D} = \frac{\sigma_1 - \sigma_2}{\sigma_1}, \qquad a_{2D} = \frac{\sigma_2 - \sigma_3}{\sigma_1}, \qquad a_{3D} = \frac{\sigma_3}{\sigma_1}$

3: $E_f = -a_{1D} \ln(a_{1D}) - a_{2D} \ln(a_{2D}) - a_{3D} \ln(a_{3D})$



(a) Entropy feature values obtained by synthetically varying eigenvalues.



(b) Entropy feature values obtained from a scan collected in Scene C.

Figure 5.1: Geometrical dimensions vs entropy feature.

Gressin et al. (2013) use this entropy feature for point selection, albeit the work has what appears to be a minor error in logic that is worth correcting. Gressin et al. (2013) assert that the larger E_f, the more prominent a single entropy dimension is. They select points with E_f greater than a specified threshold. The error in logic is revealed by Fig. 5.1 which shows the tensor-based dimensions (a_{1D}, a_{2D} and a_{3D}) against the value of the entropy feature as coloured contours. Both simulated and real data are

shown. When one entropy dimension dominates, the entropy measure is seen to decrease, with the entropy measure being largest when all the tensor dimensions are equal, suggesting the smaller E_f is indicative of a more prominent single entropy dimension.

Note that the method of Gressin et al. (2013) was included in the comparative study presented in Chapter 4 where it was found none of the point selection methods, alone or in combination, led to improved performance. This entropy concept has appeal because of its rich characterization of data. The remainder of this chapter explores an alternative entropy idea for its potential to add robustness to scan matching across different point clouds.

5.5 Eigentropy - an entropy based feature

The idea of applying an entropy-like measure to point clouds has appeal as a means for describing the local geometric structure of points in their immediate neighbourhood.

The concept of entropy has its foundations in physics where it serves as a thermodynamic quantity representing the unavailability of a system's thermal energy for conversion into mechanical work. This can be interpreted as the degree of disorder or randomness in the system. The second law of thermodynamics says that entropy always increases with time. Shannon (1948) used a logarithmic measure he termed entropy, analogous to thermodynamic entropy, to characterize the rate of transfer of information in a particular message or language, and in so doing established the field of information theory. Like thermodynamic entropy an increase in information entropy corresponds to an increase in the disorder of the message. This use is consistent with the use of entropy in general language where it broadly means the lack of order or predictability. An entropy measure for point clouds should share this characteristic: the more structured a point distribution is the lower its entropy and *vice versa*.

Equation 5.1 gives the definition for discrete entropy where X is a discrete random variable and $p(x_i)$ is the probability of the instance x_i of the variable X.

$$H(X) = -\sum_i p(x_i) \log p(x_i) \tag{5.1}$$

Several studies have used entropy approaches for working with point cloud data. Sáez et al. (2006) propose a global entropy of a point cloud, using the variable x_i to describe the number of points contained in a 3D cell and taking probability to be the ratio of points in the cell to the total number of points in the point cloud. This entropy measure gives a quantification of how points are distributed within the point cloud.

Tsin and Kanade (2004) present an alternative global entropy approach based on the Renyi's Quadratic Entropy using the variable x_i to describe the position of points and as density distribution Gaussian kernels. A point cloud with minimum entropy is one with the maximum affinity (that is, minimum

distance) between all pairs of points.

Kadir and Brady (2001) use a local entropy measure based on the grey scale of pixels in a neighbourhood of points in a computer image. The variable x_i is the 8 bit grey level descriptor and the probability is found from the ratio of the repetition of that descriptors to the total number of descriptor present in the neighbourhood.

Wu and Wang (2007) introduce a local entropy based on the mean curvature obtained in the neighbourhood of a query point. The discrete variable x_i is the mean curvature and the probability is determined from a normalized version of the point mean curvatures over the neighbourhood.

Fiolka et al. (2012) introduce a local entropy based on the angle of surface normals obtained in the neighbourhood of a query point. They use an orientation histogram to count the occurrences of normal orientation into a defined angle bin. The number of occurrences in each angle bin defines the discrete variable x_i and the probability is computed from the normalized histogram of these occurrences.

The methods presented in Wu and Wang (2007) and Fiolka et al. (2012) share the same idea of geometric disorder of information: the greater the value of the entropy, the greater the geometric disorder. In Wu and Wang (2007), interest is focussed on variation of surface curvature; in Fiolka et al. (2012) interest is focussed on disorder of the surface normals. However both methods struggle when the points are lying in a perfect plane or along a straight line and thus the orientation of the normals are 90°. Neither feature can distinguish between those geometries.

The entropy feature introduced in Section 5.4 incorporates tensor terms to obtain a quantification of the geometrical features (line, plane and ball) of the neighbourhood of a point. This entropy feature gives a relationship for the dimensionality features obtained from the neighbourhood of points. However, Fig. 5.1 shows that the entropy feature doesn't match with the definition of entropy for geometric disorder: the data is not necessarily well structured and both highly ordered (e.g. a line) and widely scattered distributions can have low entropy.

The entropy measure, E_g, defined in Algorithm 4, describes the geometric structure of a point cloud and has the property of increasing entropy with decreasing structure. The name given to this entropy-like measure is *eigentropy* a portmanteau of eigen and entropy. The calculation of eigentropy is similar to the eigenentropy measure given in West et al. (2004). The main idea of the eigentropy metric is to exploit the relationship of eigenvalues with the spacial distribution of the underlying points as summarized in Table 5.2.

Algorithm 4 Eigentropy procedure.

Require: Eigenvalues, λ_i, obtained by PCA computed from a given neighbourhood. σ_i defines the standard deviation of the eigenvector $\hat{\mathbf{e}}_i$. $\hat{\sigma}_i$ represent the normalized standard deviation.

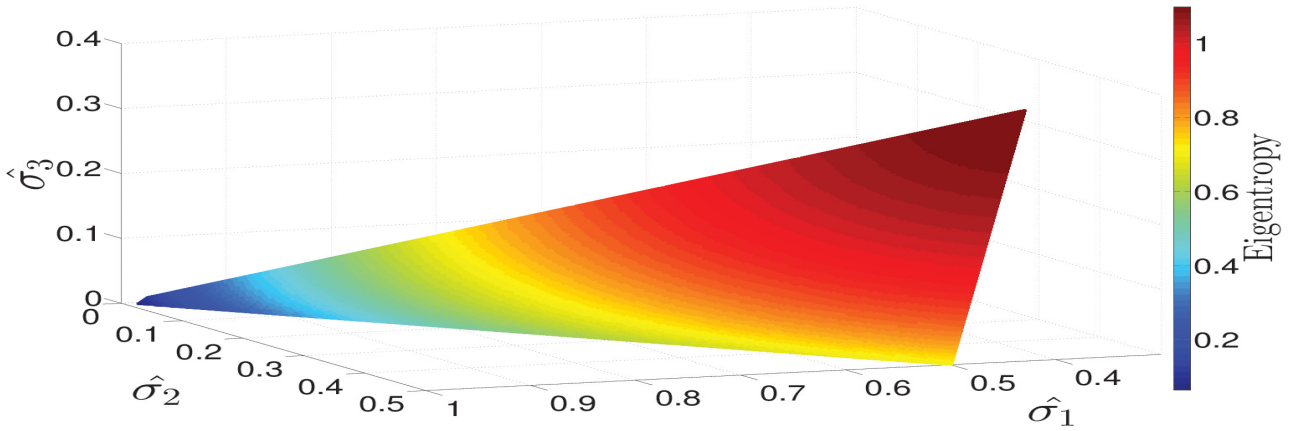
Ensure: E_g defines the eigentropy feature.

1: $\sigma_i = \sqrt{\lambda_i}$

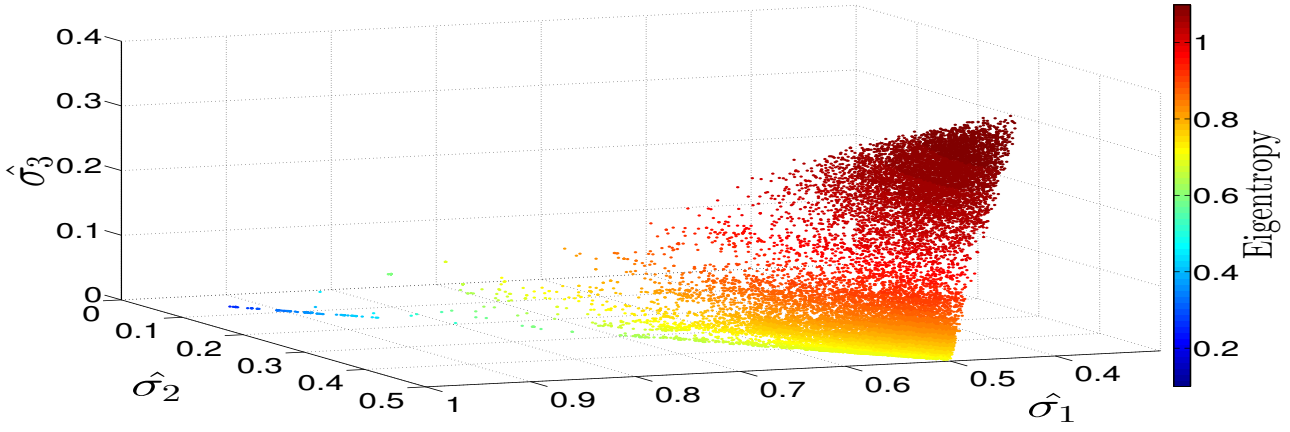
2: $\hat{\sigma_1} = \frac{\sigma_1}{\sigma_1+\sigma_2+\sigma_3}, \quad \hat{\sigma_2} = \frac{\sigma_2}{\sigma_1+\sigma_2+\sigma_3}, \quad \hat{\sigma_3} = \frac{\sigma_3}{\sigma_1+\sigma_2+\sigma_3}$

3: $E_g = -\hat{\sigma_1} \ln(\hat{\sigma_1}) - \hat{\sigma_2} \ln(\hat{\sigma_2}) - \hat{\sigma_3} \ln(\hat{\sigma_3})$

Figure 5.2 shows the relation of eigenvalues and eigentropy. This relation shows that when $\hat{\sigma_1} \simeq \hat{\sigma_2} \simeq \hat{\sigma_3}$ (scattered geometrical behaviour) the eigentropy is larger (dark red region). When one eigenvalue dominates, corresponding to points distributed in a line, eigentropy is small (dark blue points).



(a) Eigentropy values obtained by synthetically varying eigenvalues.



(b) Eigentropy values obtained from a scan collected in Scene C.

Figure 5.2: Eigenvalues vs eigentropy.

Point cloud for Scenes A, B and C (Chapter 3) with points coloured by eigentropy are shown in Figs. 5.3, 5.4 and 5.5 respectively. The eigentropy value quantifies the geometric disorder or information of the neighbourhood of a point. Among other things, thresholding eigentropy provides a means for selecting points that occur either on smooth surfaces or in straight lines.

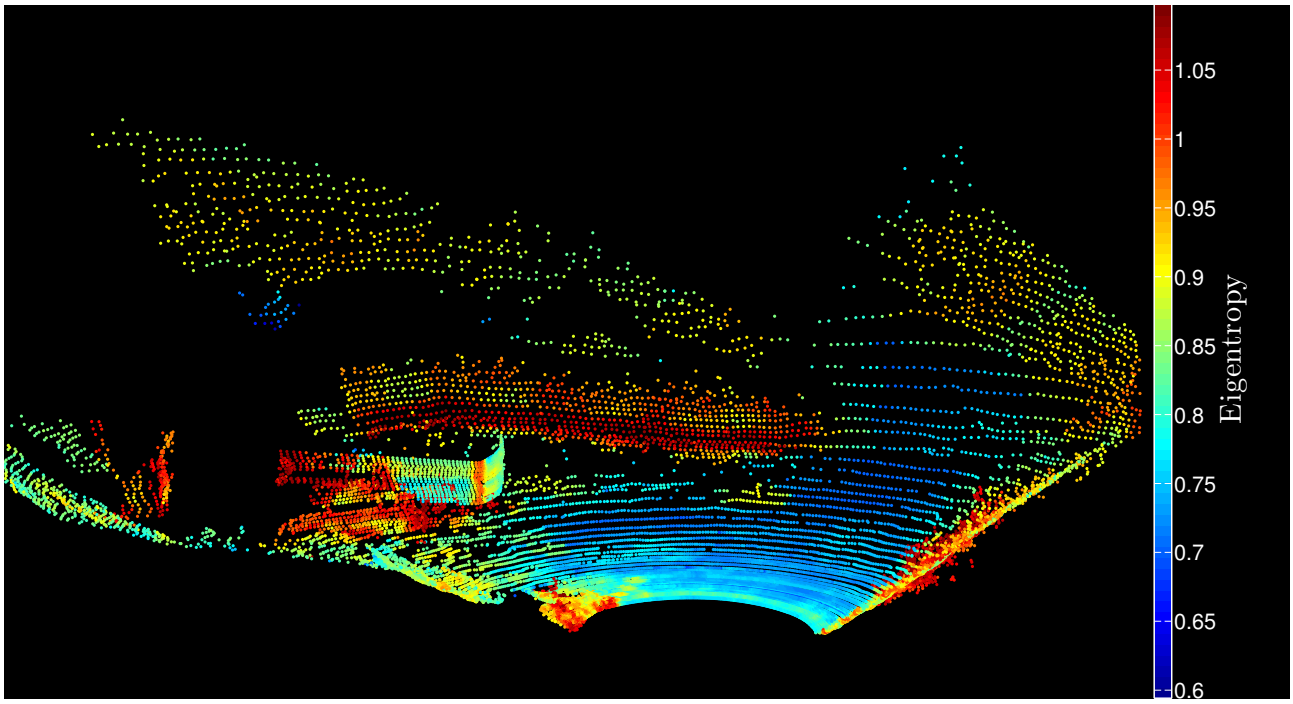


Figure 5.3: Eigentropy for a scan collected in Scene A.

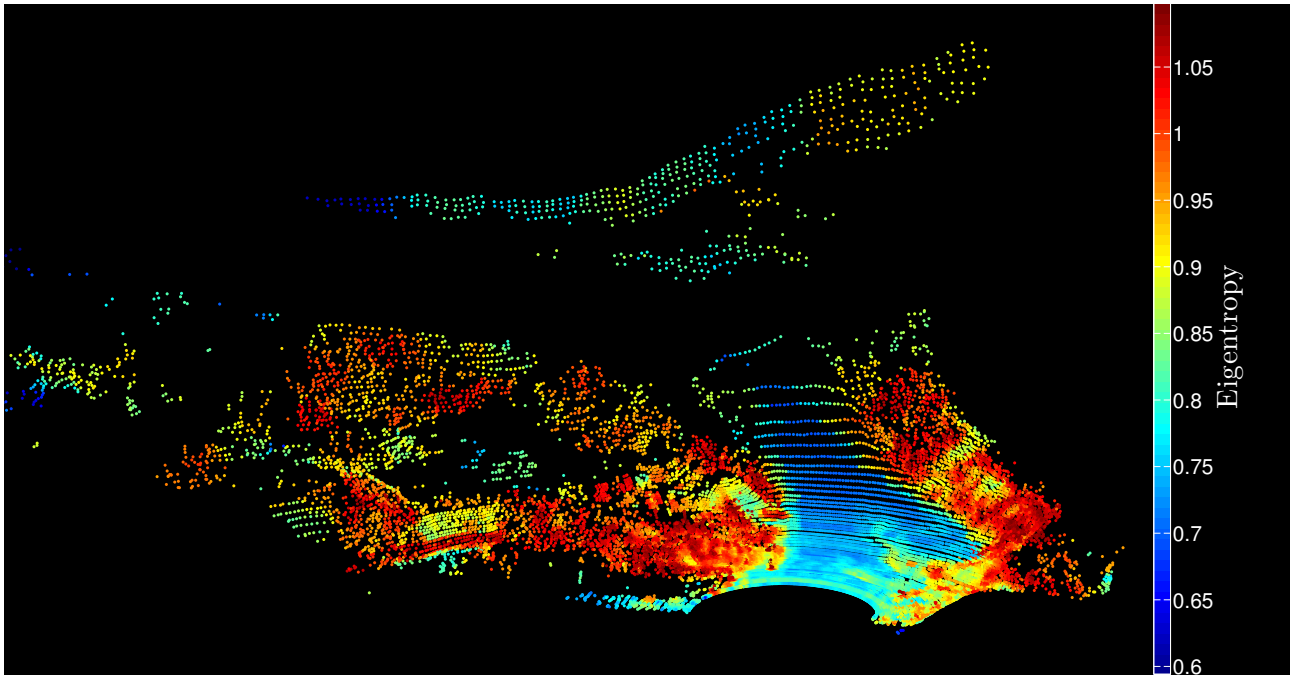


Figure 5.4: Eigentropy for a scan collected in Scene B.

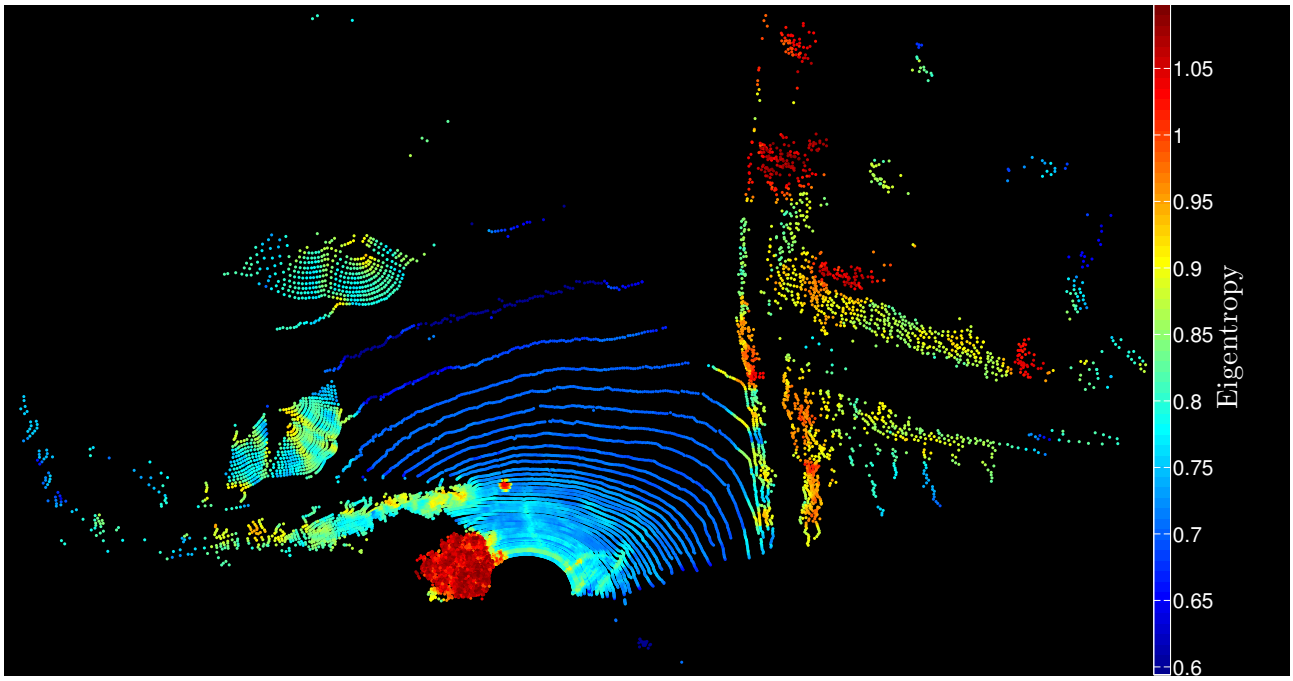


Figure 5.5: Eigentropy for a scan collected in Scene C.

The eigentropy measure computed by Algorithm 4 is intended as a local geometric entropy, however this concept can be expanded to quantify the geometric information of the whole point cloud by taking the average of the point eigentropy. This is the total eigentropy of the point cloud. Equation 5.2 shows the total eigentropy (H_g), where $E_g(\mathbf{p}_i)$ is the eigentropy of the point $\mathbf{p}_i$ of the point cloud and N is the cardinality of the point cloud,

$$H_g(\mathcal{P}^N) = \frac{1}{N}\sum_{i=1}^{N} E_g(\mathbf{p}_i). \tag{5.2}$$

Two methods that use eigentropy have been developed in this work. These methods are termed the *eigentropy filter* and *unilateral eigentropy rejection* which are applied as part of the *point selection* and *pair rejection* stages of the ICP process respectively.

5.5.1 Method 1: The eigentropy filter

The first of the three new methods introduced in this chapter is the eigentropy filter used for point selection. The eigentropy filter retains points in structured regions and rejects points in regions that are disordered by thresholding against K_f. Figure 5.6 shows the eigentropy coloured point cloud of Fig. 5.3, Scene A, with points above $K_f = 0.98$ removed. The points removed correspond to vegetation in the scene. The point cloud is seen to have a visibly crisper structure when points above this threshold are removed.

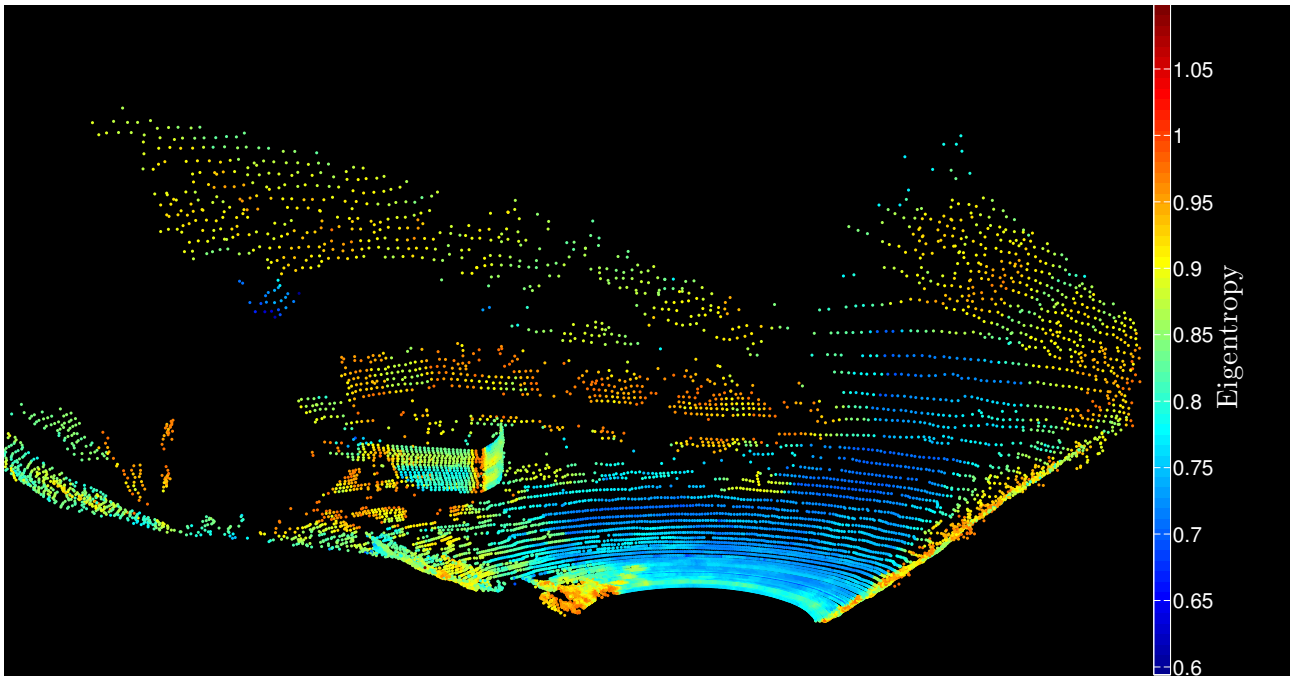


Figure 5.6: Filtered point cloud by eigentropy filter with $E_g > 0.98$, for Scene A.

The threshold value, K_f, of the eigentropy filter for effective point selection varies between different point clouds. Figure 5.7 shows, for the point clouds of data across the three scenes (297 points clouds in total) of the thesis data set, the total eigentropy and the value of K_f determined to give the most accurate scan match for each point cloud. Figure 5.7 shows the relation of the total eigentropy of 297 scans (point clouds) and the value of threshold, K_f, determined to give the most accurate scan match for each scan.

Several points are worth noting. First, the total eigentropy is clustered by scene. Considering the total eigentropy to be a measure of complexity in the point cloud, provides an observation that is consistent with expectation: Scene B is more complex than Scene A which is more complex than Scene C.

The range of total entropy for each cluster also varies with each scene: C is the most dispersed and A is the least. It was noted in Chapter 4 that scan matches for Scene C were less accurate, in general than those from, Scenes A and B and this supports (but does not prove) the hypothesis that the quality of scan matching is connected with entropy.

The values of K_f for most the accurate fitting of each scan match fall in the range $0.85 < K_f < 1.1$ and appear to be loosely correlated to H_g. For example, centres of the clusters representing Scences A and B have lower K_f than that of Scene C. This relation is amplified by the average value of K_f in bins 0.0033 units wide. These averages are shown by red dots in Fig. 5.7, and are the average values of the optimal values of K_f. Also shown is the line of best fit through them.

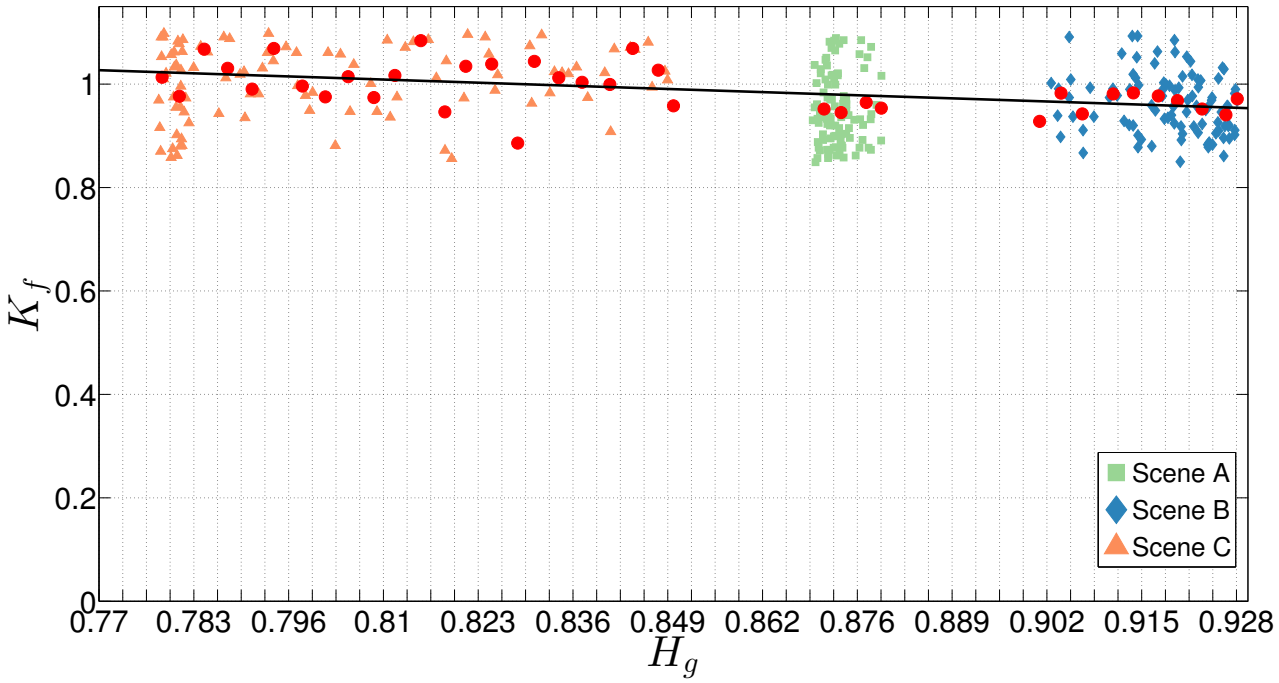


Figure 5.7: Relation of eigentropy filter threshold, K_f, and the total eigentropy, H_g, for the most accurate single scan matching across the three scenes of the variant g4w0α1.

The eigentropy filter uses this line to determine the value of K_f to be applied. This dynamic selection of K_f based on H_g is intended to capture the general nature of the correlation observed, namely the higher the value of total entropy the lower the value of K_f for most accurate scan matching. Using this approach to dynamically selected K_f has been found to give better overall scan matching than a fixed K_f. The eigentropy filter method is shown in the Algorithm 5.

Algorithm 5 Eigentropy filter.

Require: $E_g(\mathbf{p}_i)$ is the eigentropy of the point, $\mathbf{p}_i$. H_g is the total eigentropy of the point cloud $\mathcal{P}^n$. a and b are the coefficients of the linear relation of the threshold K_f with the total eigentropy.

Ensure: $N \leq n$

1: $H_g \leftarrow H_g(\mathcal{P}^n)$
2: $K_f \leftarrow -aH_g + b$
3: **for all** $\mathbf{p}_i \in \mathcal{P}^n$ **do**
4: **if** $E_g(\mathbf{p}_i) \leq K_f$ **then**
5: $\mathcal{P}^N \leftarrow \mathbf{p}_i$ (retain)
6: **else**
7: $\mathcal{P}^N \nleftarrow \mathbf{p}_i$ (filter)
8: **end if**
9: **end for**

5.5.2 Method 2: unilateral eigentropy rejection

The eigentropy rejection is defined as the rejection of paired points whose combined entropy is greater than a given threshold. Equation 5.3 defines the eigentropy of paired points $E_p\left(\langle \mathbf{p}, \mathbf{q} \rangle_i\right)$, where $\langle \mathbf{p}, \mathbf{q} \rangle_i$ is the correspondence of a point, $\mathbf{p}_i$, of the input point cloud to a point, $\mathbf{q}_j$, of the reference point cloud.

$$E_p(\langle \mathbf{p}, \mathbf{q} \rangle_i) = w_p E_g(\mathbf{p}_i) + w_q E_g(\mathbf{q}_i). \tag{5.3}$$

The weighting factors, w_p and w_q, assign relevance to the eigentropy of the input and reference points respectively. Unilateral eigentropy rejection is defined for exclusively prevalence of eigentropy of the reference point in the Eqn. 5.3, i.e. $w_p = 0$ and $w_q = 1$.

The unilateral rejection method aims to preserve pairs with accurate surface normals, and this is expected to be particularly beneficial for improving the accuracy of the minimization step using the point-to-plane method. Figure 5.8 shows the resulting input point cloud, for Scene A, after applying the unilateral eigentropy rejection with the threshold set at 0.86.

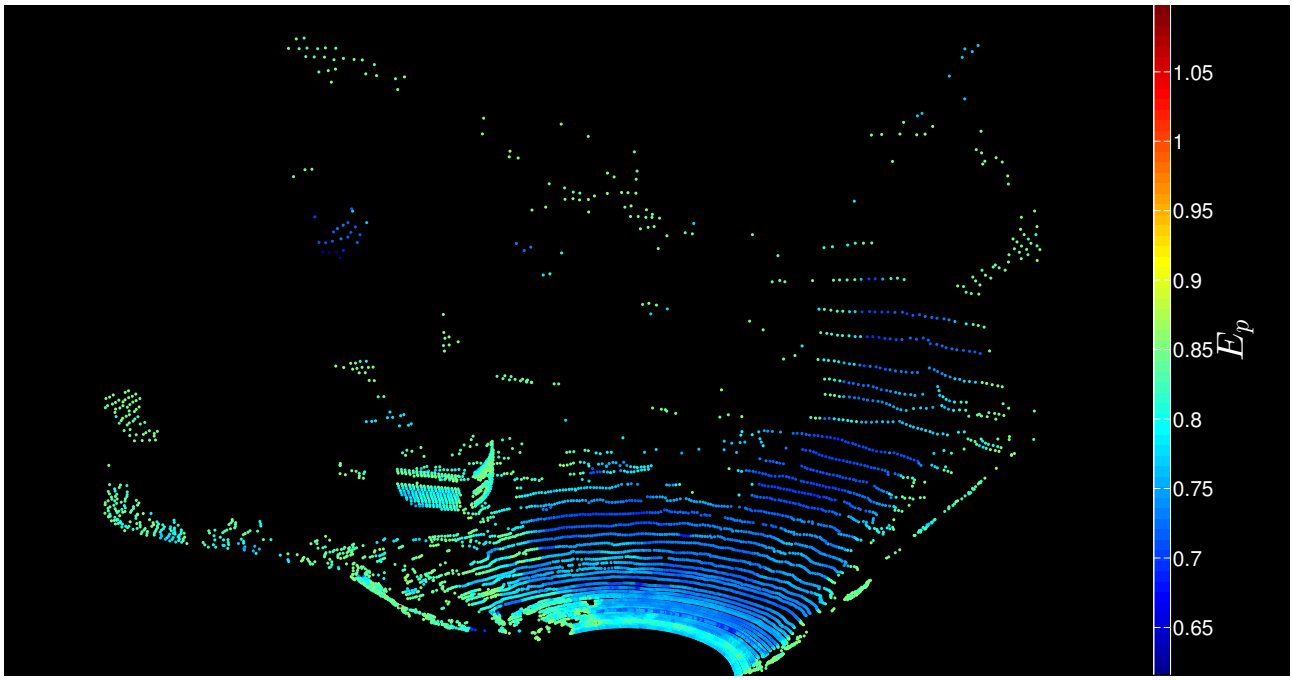


Figure 5.8: Eigentropy of the paired points, rejecting $E_p > 0.86$, for Scene A.

The unilateral eigentropy rejection method is described in Algorithm 6.

Algorithm 6 Unilateral eigentropy rejection.

Require: $E_p(\langle \mathbf{p}, \mathbf{q} \rangle_i)$ is the eigentropy of paired points $\langle \mathbf{p}, \mathbf{q} \rangle_i$. $\mathcal{G}$ is the set of paired points. w_p and w_q are the weights of the eigentropy of the input and reference point cloud, respectively. K_r is the rejection threshold.

Ensure: $N' \leq N$

```
w_p ← 0
w_q ← 1
K_r ← 0.86
for all ⟨p, q⟩_i ∈ G^N do
    if E_p(⟨p, q⟩_i) ≤ K_r then
        G^N' ← ⟨p, q⟩_i   (retain)
    else
        G^N' ↚ ⟨p, q⟩_i   (reject)
    end if
end for
```

The threshold, K_r, is set to 0.86 for the three studied scenes.

5.6 Method 3: point matching by normal deviation

The third new method is not entropy based but has proven effective when applied to point matching.

Finding correspondence between points of the input and reference point cloud is of crucial importance for the convergence of ICP (Rusinkiewicz and Levoy, 2001). The ICP algorithm as presented by Besl and McKay (1992) introduced the nearest neighbour matching that simply found closest points of the reference point cloud to every point in the input point cloud. This method is particularly efficient to implement when *kd*-trees are used for nearest neighbour lookup. This is still the most common point matching method for fine registration.

Another approach for point matching consists of finding correspondence between features (see Section 5.2), mainly because some features are distinctive throughout the point cloud. Considering this, others have introduced matching methods by using rich descriptors for coarse registration (Rusu et al., 2009; Brusco et al., 2005; Diez et al., 2012; Yao et al., 2010). These methods are good for finding an initial alignment when a large misalignment is present, however, they are not suitable for fine registration.

Point matching methods based on features for fine registration are addressed as an enhanced version of the classical nearest neighbour method. This means adding features to the search for correspondence by augmenting the searching space. Examples of these methods are in Feldmar et al. (1995); Sharp

et al. (2002); Skotheim et al. (2012); Schutz et al. (1998) and Akca (2005). Chapter 4 shows that nearest neighbour enhanced by surface normals produce more accurate registration than the vanilla nearest neighbour matching. However, using normals exclusively or any other low dimension feature for point matching is virtually impossible because of the repetition of the value of these features across the point cloud.

The main problem of registering consecutive scans obtained from a moving sensor is that a direct point correspondence between consecutive scans does not exist. Strictly speaking, the scans were obtained from different scenes. The human brain instinctively observes scene correspondences, however a machine must interpret points which don't have a direct correspondence. Moreover, minimizing the distance between points leads to erroneous registrations, as was the case for the point-to-point minimization, see Chapter 4.

This section introduces a novel point matching method. The point matching problem is addressed with a simple but powerful method that matches points only from their normal. This method compares the angular deviation of the k nearest neighbours of the query point and is termed *matching by normal deviation*. The matching by normal deviation method has two steps: i) a k nearest neighbour search, kNN, and ii) an angular search. Algorithm 7 describes the matching by normal deviation method.

Algorithm 7 Matching by normal deviation procedure.

Require: kNNsearch is a k nearest neighbour search algorithm . $\mathcal{Q}^k$ is the set of k nearest points to $\mathbf{p}_i$. $\mathbf{n}_{p_i}$ and $\mathbf{n}_{q_k}$ are the normal vectors associated with $\mathbf{p}_i$ and $\mathbf{q}_k$, respectively. θ_k is the angular deviation of the normal $\mathbf{n}_{q_k}$to the normal $\mathbf{n}_{p_i}$. Threshold angle Θ_t used for matching.

Ensure: $\langle \mathbf{p}_i, \mathbf{q}_j \rangle$ is the correspondence of the query point $\mathbf{p}_i$ on the input point cloud $\mathcal{P}^N$ with a point $\mathbf{q}_j$ on the reference point cloud $\mathcal{Q}^M$.

$k \leftarrow 10$
for all $\mathbf{p}_i \in \mathcal{P}^N$ **do**
  $\mathcal{Q}^k \leftarrow \texttt{kNNsearch}(\mathbf{p}_i, \mathcal{Q}^M, k)$
  for all $\mathbf{q}_k \in \mathcal{Q}^k$ **do**
    $\theta_k \leftarrow \cos^{-1}(\mathbf{n}_{q_k} \cdot \mathbf{n}_{p_i})$
  end for
  $\theta_j \leftarrow \min(\theta_1, \ldots, \theta_k)$
  if $\theta_j > \Theta_t$ **then**
    $\mathbf{q}_j \leftarrow \min(\mathbf{q}_1, \ldots, \mathbf{q}_k)$
  end if
  $\langle \mathbf{p}_i, \mathbf{q}_j \rangle \leftarrow$ matching
end for

The number of nearest neighbours is set to $k = 10$, through trial and error, for the three studied

scenes.

5.7 Evaluation

The evaluation of these three new methods follows the methodology used in Chapter 4. The three data sets and the three performance metrics (accuracy, precision and computational efficiency), remain the same. See Chapter 3 for details.

The addition of these new methods increases the number of ICP variants from 20,736 to 73,728. As in Chapter 4, ICP variants are identified by an alphanumeric code, where each letter/number represents a method applied at different stages.

Table 5.3 shows the set of methods evaluated for this work. The new methods presented are applied to *point selection*, *point matching* and *pair rejection*.

Table 5.3: Selected strategies for performance comparison.

Stage	**Code**	**Method**	**Ref.**
Points selection	a	All points	Besl and McKay (1992)
	b	Outliers removal filter	Xie et al. (2004)
	c	Density filter	Lalonde et al. (2006)
	d	Geometrically stable sampling	Gelfand et al. (2003)
	e	Entropy feature filter	Gressin et al. (2012)
	f	Dimensionality based selection	Gressin et al. (2012)
	g	**Eigentropy filter**	**This work**.
Neighbourhood selection	1	Constant	Rusinkiewicz and Levoy (2001)
	2	Entropy feature minimization	Demantké et al. (2011)
	3	Density adaptation	Wiemann et al. (2010)
	4	Bounded radius	Lalonde et al. (2005); Mitra et al. (2004)
Point matching	z	NN enhanced by normals	Feldmar et al. (1995)
	y	NN enhanced by moment invariants	Sharp et al. (2002)
	x	Nearest neighbours (NN)	Besl and McKay (1992)
	w	**Normal deviation**	**This work**.
Weighting	0	Constant	Besl and McKay (1992)

Continued on next page

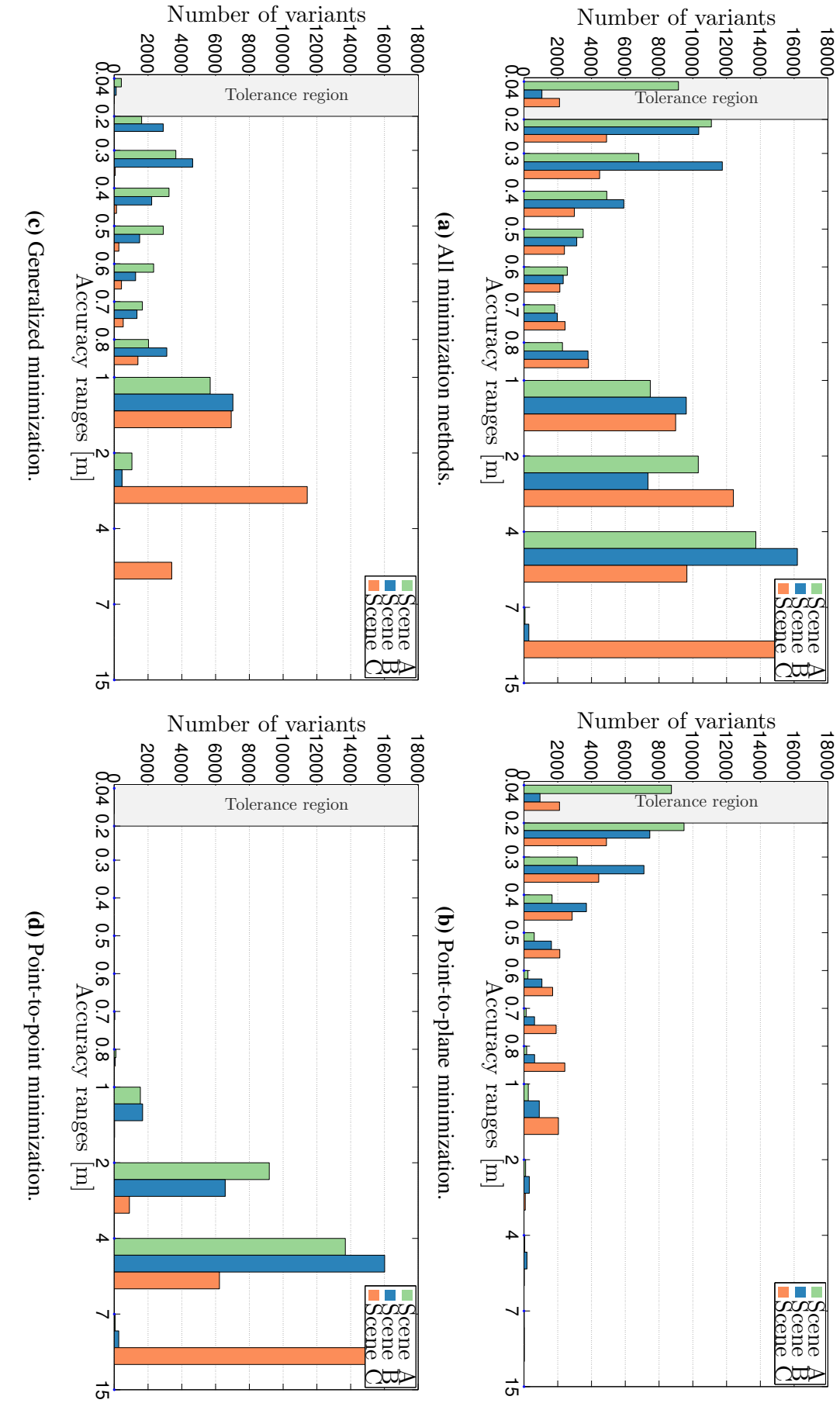
Table 5.3 – *Continued from previous page*

Stage	**Code**	**Method**	**Ref.**
	1	Distance	Godin et al. (1994); Rusinkiewicz and Levoy (2001)
	2	Normals compatibility	Rusinkiewicz and Levoy (2001)
Rejection	α	No rejection	Zhang (1994)
	β	Distance by worst percentage	Rusinkiewicz and Levoy (2001); Pulli (1999)
	γ	Angular deviation	Zhang (1994)
	δ	Adaptive distance by variance	Zhang (1994)
	$\boldsymbol{\epsilon}$	**Unilateral eigentropy**	**This work**.
Minimization	0	Point-to-point	Besl and McKay (1992)
	1	Point-to-plane	Chen and Medioni (1992)
	2	Generalized	Segal et al. (2009)

5.8 Results and observations

The aim of this section is to evaluate the relative performance of the three novel methods presented. Figure 5.9 shows accuracy histograms by minimization methods, coloured by scene. The distribution of variants throughout the whole accuracy range is similar to that presented in Fig. 4.5. However, the lower bound of the accuracy range is lower (0.04 m verse 0.08 m) indicating some ICP variants using these novel methods have improved accuracy. Note also that some new variants are less accurate, e.g. worse case accuracy for point-to-plane minimization in Fig. 4.5 was 3 m; and in Fig. 5.9 is 7 m.

Figures 5.10, 5.11 and 5.12 provide relationship graphs for the most accurate, precise, and computational efficient variants. A variant is considered accurate if its accuracy after 100 scans is below 0.2 m; 0.266 % of the 73,728 variants fit this criteria. A variant is considered precise if its precision is less that 0.1 m over 100 scans; 28.7 % of variants are precise. A variant is considered computationally efficient if its relative computation time is no more than 3 times the fastest variant; 6.78 % of variants meet this criteria. The interpretation of these graphs is the same as observed in Chapter 4. Each method at each stage, alone or in combination, is represented by an alpha numeric code as per



(a) All minimization methods.

(b) Point-to-plane minimization.

(c) Generalized minimization.

(d) Point-to-point minimization.

Figure 5.9: Accuracy histograms for the scenes by minimization method.

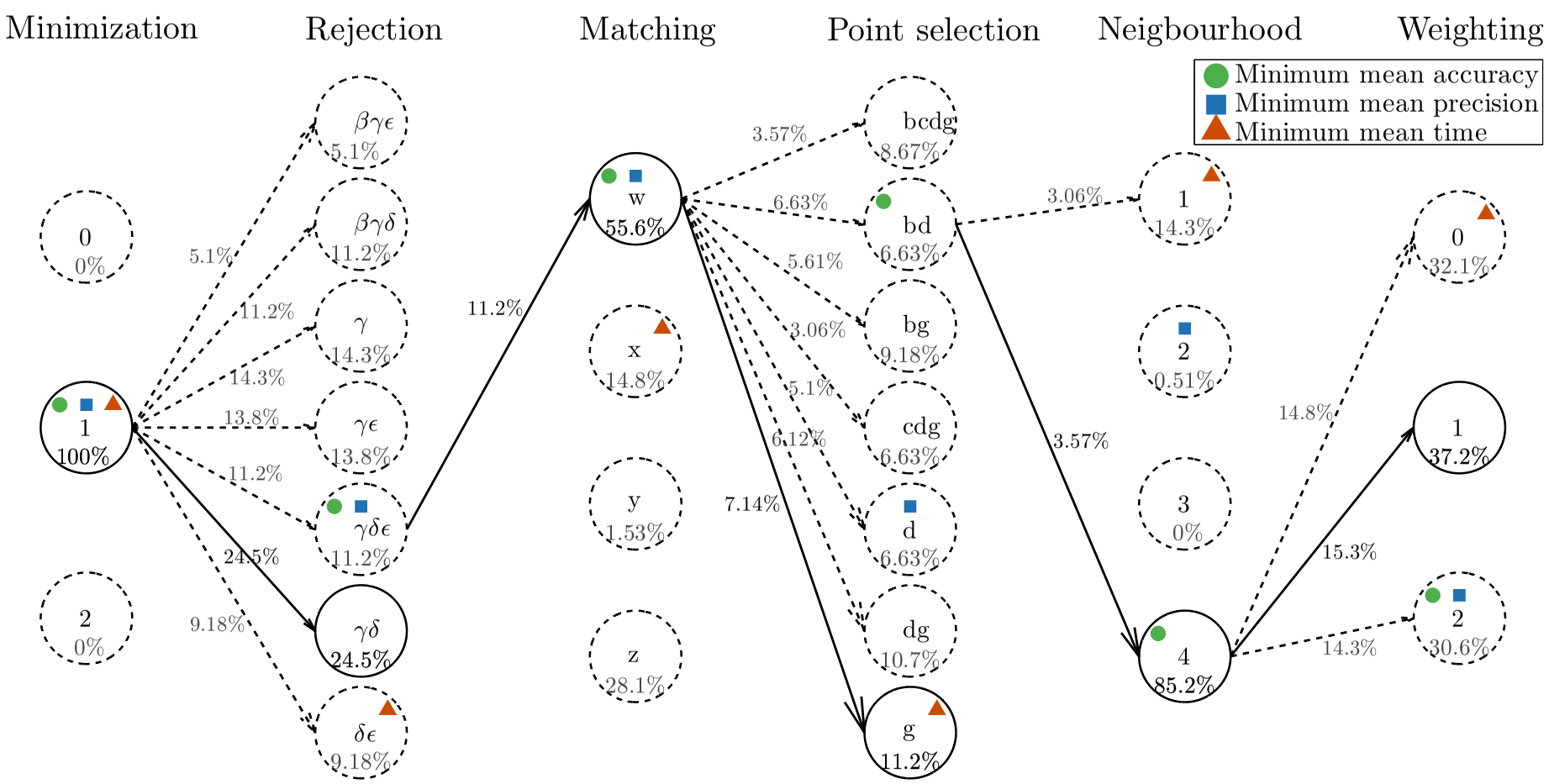
Table 5.3. Note that only the seven most common variants are listed for rejection and minimization stages.

The new variants that make use of one or more of the three methods introduced in this chapter add to those identified in Chapter 4 and alter the population demographics. Some things are unchanged. As in Chapter 4, there is strong preference for point-to-plane minimization with all (100 %) of the accurate variants, 74.7 % of the precise, and 51.7 % of the efficient variants using this method.

Some of the changes are more pronounced. For example 100 % of the accurate variants in Chapter 4 used the bounded radius (Method 4) for neighbourhood selection, however, among the broader set of variants, constant neighbourhood (Method 1) appears in 14.3 % of the accurate variants and entropy feature minimization (Method 2) in 0.51 %.

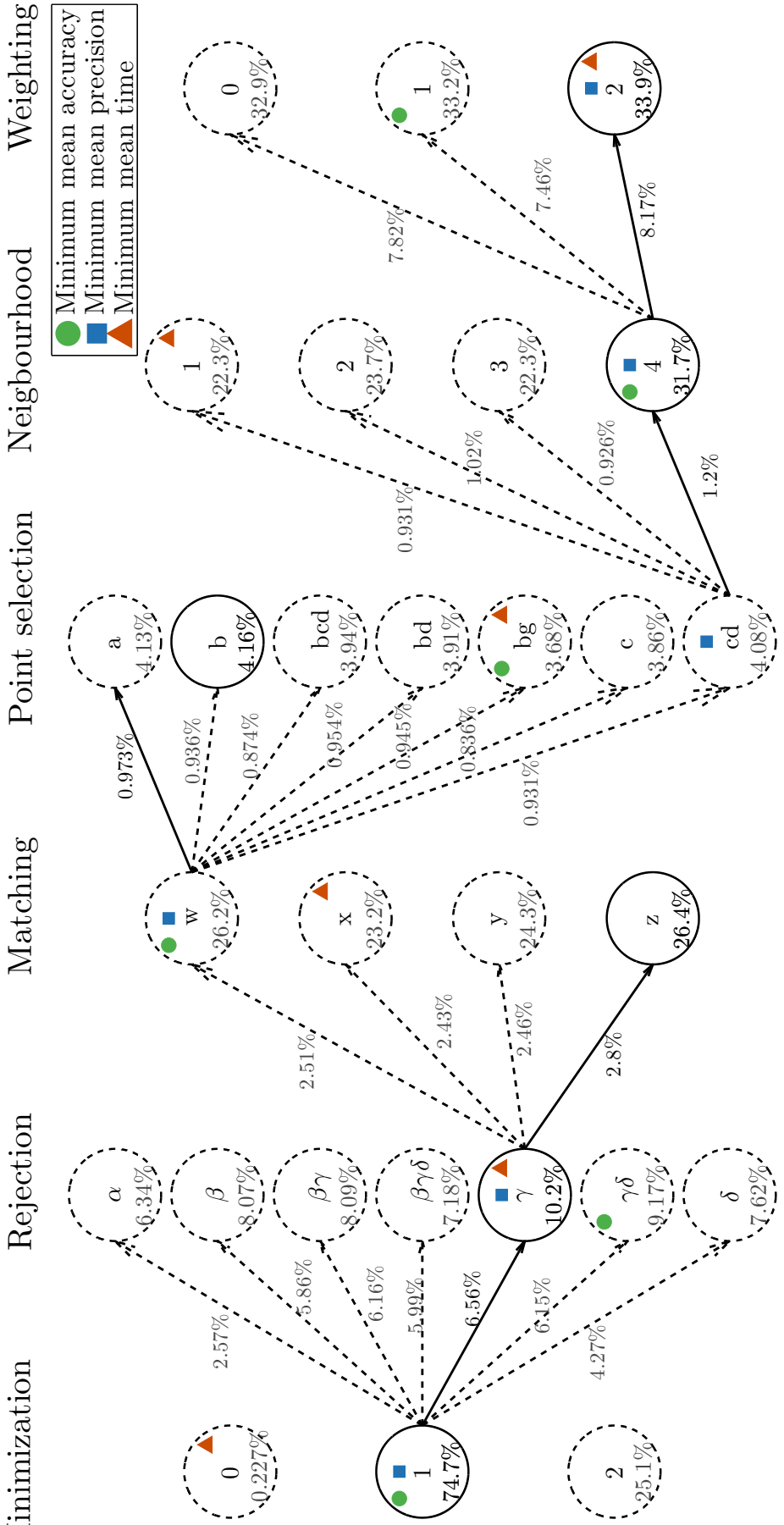
Figure 5.11 shows a similar occurrence distribution of the most precise variants compared with the results in Chapter 4. Similarly Fig. 5.12 shows that the new variants set has little impact on the computational efficiency.

The influence of each new method is now considered in turn against the population set.



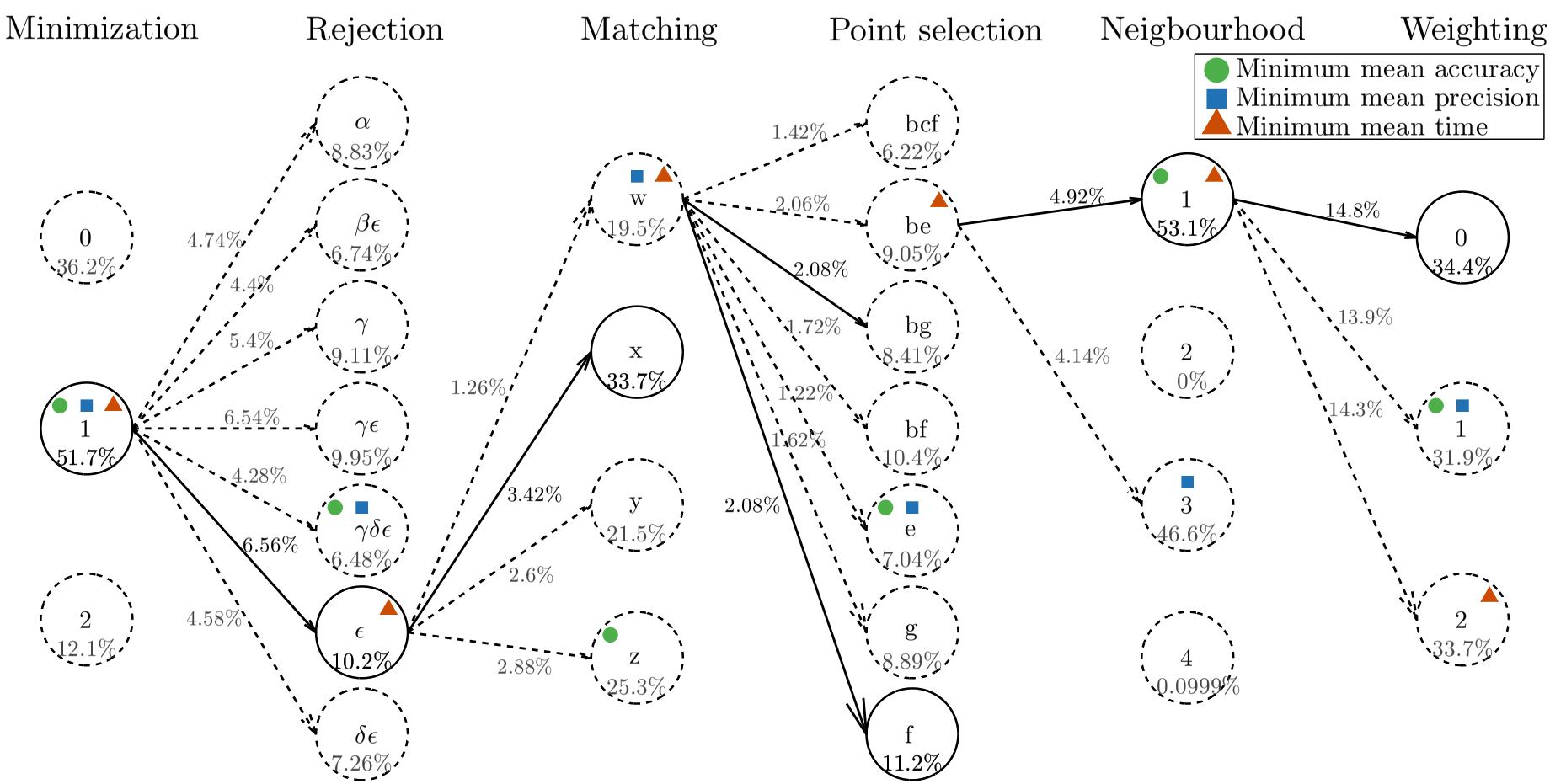
The 0.266% of 73,728 ICP variants are within the accuracy tolerance for the three scenes ($\nu \leq 0.2\,m$).

Figure 5.10: Method relationship diagram within the accuracy tolerance.



The 28.7% of 73,728 ICP variants are within the precision tolerance for the three scenes ($\rho \leq 0.1\,m$).

Figure 5.11: Method relationship diagram within the precision tolerance.



The 6.78% of 73,728 ICP variants are within the computational efficiency tolerance for the three scenes ($rct \leq 3$).

Figure 5.12: Method relationship diagram within the relative computational time tolerance.

5.8.1 Eigentropy filter

The eigentropy filter is used for point selection. Thirty-two point selection filters have been considered, using the methods alone or in combination. Figure 5.13 shows the frequency of occurrence of accurate variants and interestingly the eigentropy filter (Method g) is the most prevalent of these, either along alone or in combination with other methods. Figure 5.14 shows that the average accuracy of the most accurate variants that simultaneously meet this performance metric in three studied scenes. The best average accuracy is achieved by outlier removal (Method b) in combination with geometrically stable sampling (Method d), however methods a, b, d, e, f, and g alone have similar performance.

From Fig. 5.11 shows that outlier removal (Method b) represents 4.16 % of the most precise variants.

The overall conclusion drawn here is that whilst the average performance of variants is similar for every point selection method, eigentropy filtering is more prevalent among the accurate variants and on this basis might be considered preferable.

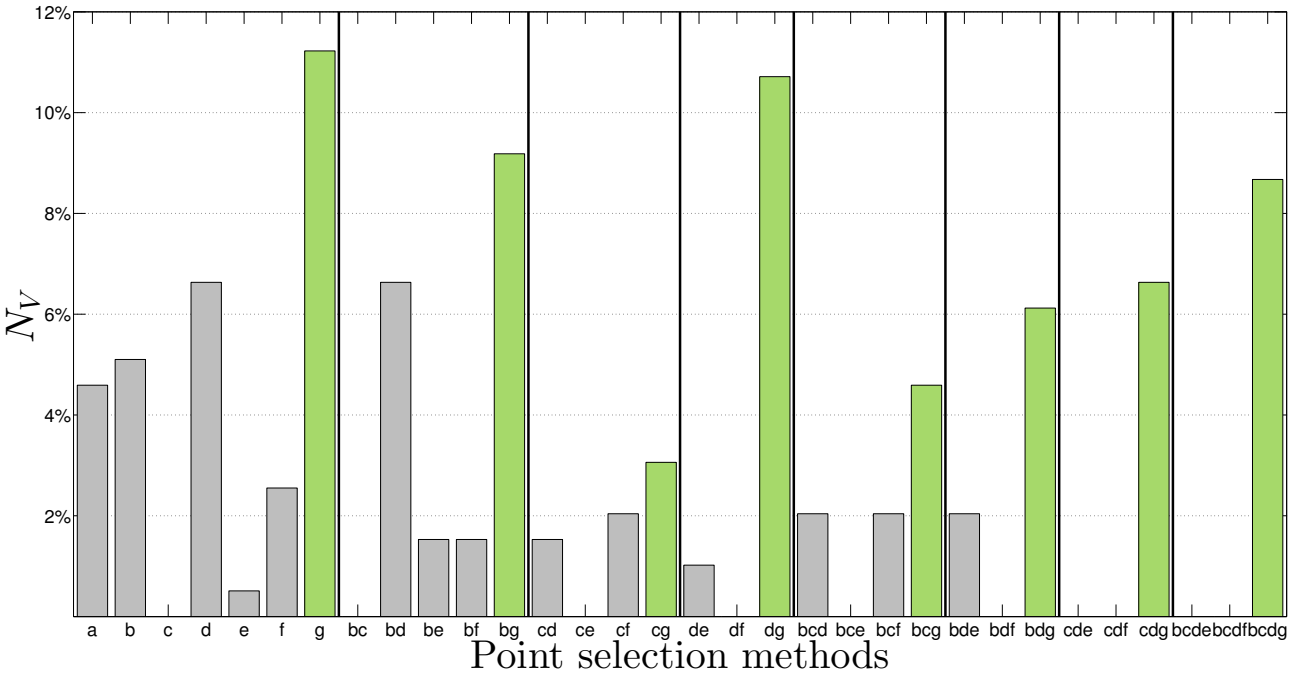


Figure 5.13: Occurrence, N_V, of point selection methods within the accuracy tolerance. Bars coloured green are methods that use the eigentropy filter, Method g.

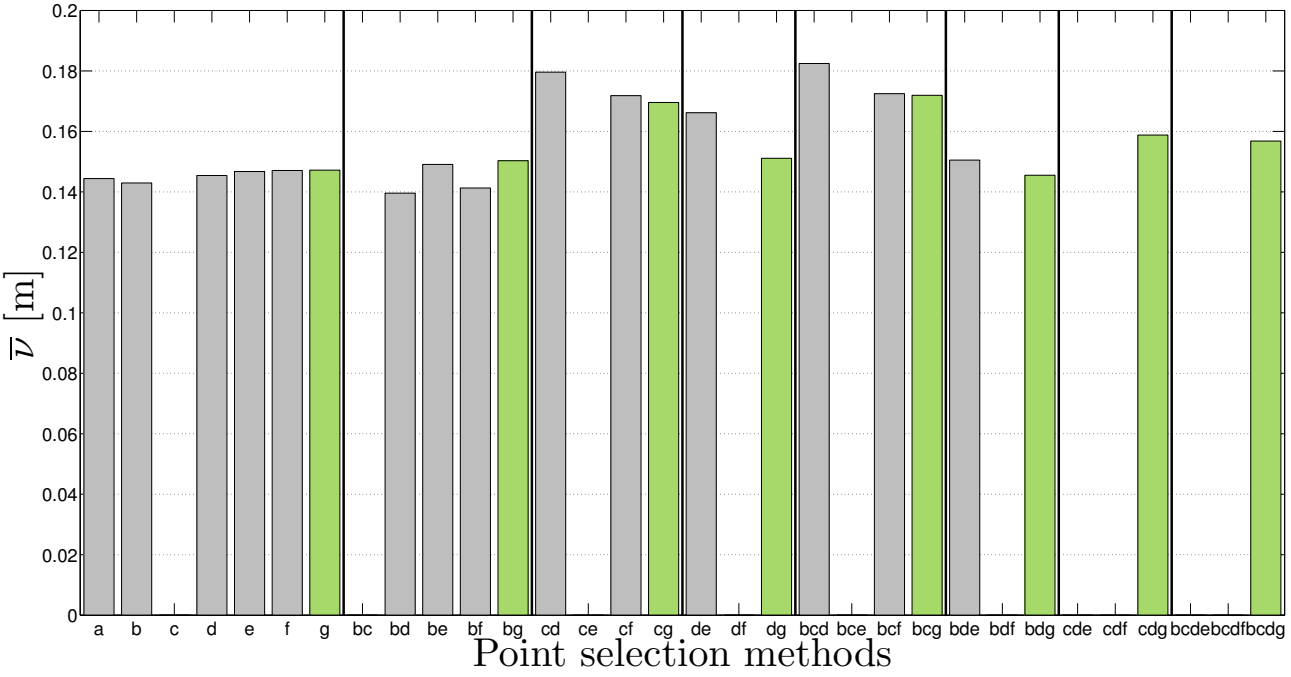


Figure 5.14: Mean accuracy, $\overline{\nu}$, for point selection methods. Bars coloured green are methods that use the eigentropy filter, Method g.

5.8.2 Matching by normal deviation

Figure 5.10 shows that matching by normal deviation (Method w) occurs in 55.6 % of the most accurate ICP variants. This is nearly twice as many occurrences than the nearest neighbour enhanced by normal method (Method z), which was identified in Chapter 4. Matching by normal deviation (Method w) clearly has the best average accuracy of the four methods considered for matching, from the most accurate variants, see Fig. 5.15.

Figure 5.11 shows that the four matching methods occur in the precise variant sets. The average precision of variants using matching by normal deviation (Method w) is better than nearest neighbours enhanced by moment invariants (Method y), see Fig. 5.16. Method y delivered the best average precision performance in Chapter 4.

Figure 5.12 shows that efficient variants using matching by normal deviation (Method w) had, on average, relative computation times lower than the other three variants

The conclusion drawn is that matching by normal deviation (Method w) is the preferred matching method against all three criteria.

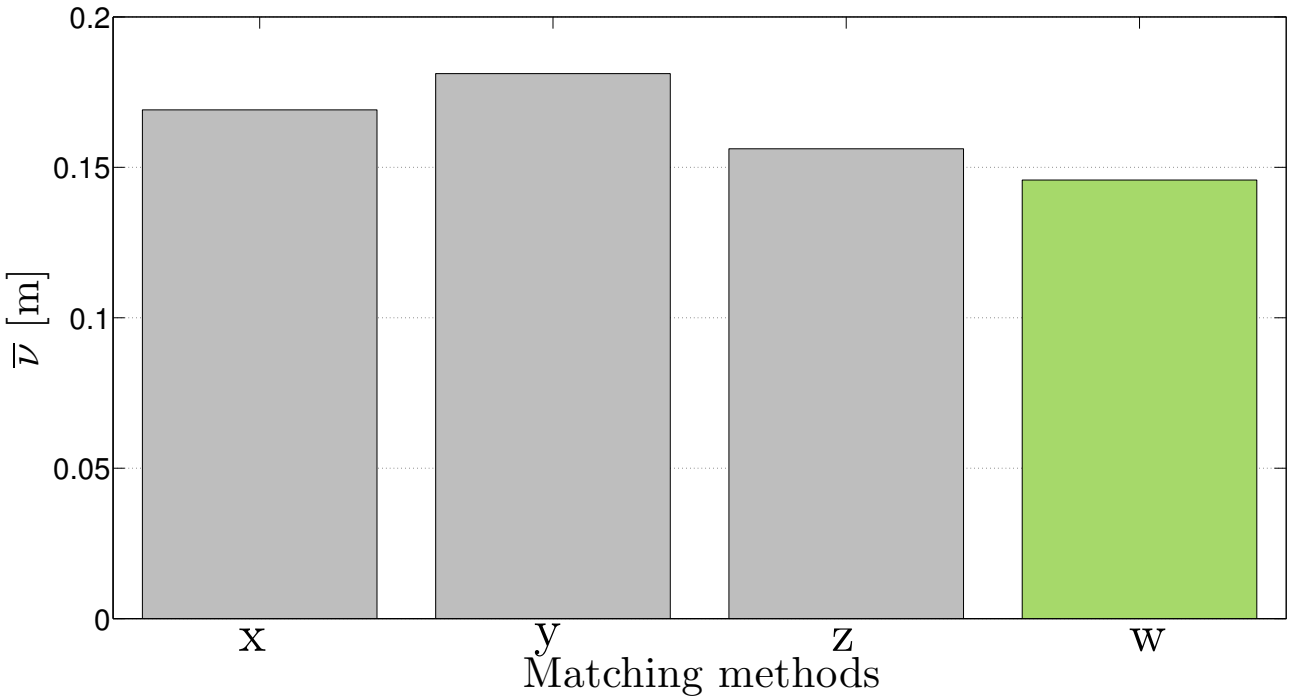


Figure 5.15: Mean accuracy, $\overline{\nu}$, for point matching methods. Bars coloured in green show the matching by normal deviation, Method w.

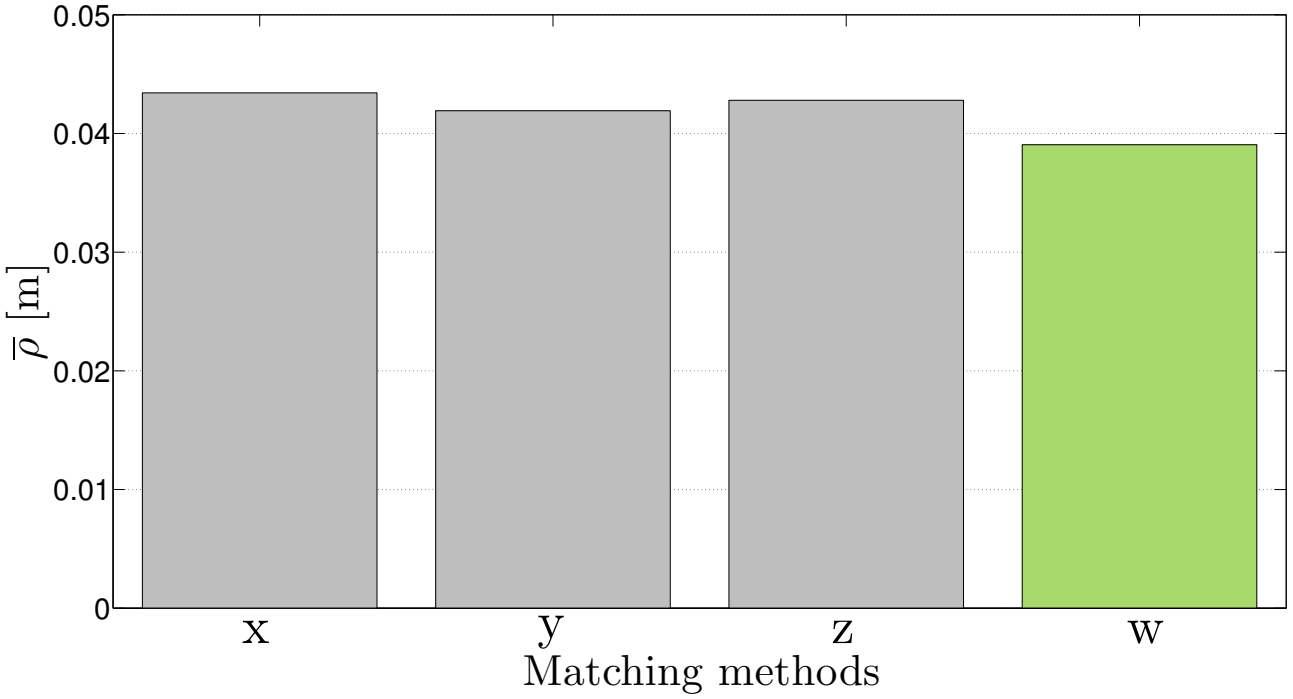


Figure 5.16: Mean precision, $\overline{\rho}$, for point matching methods. Bars coloured in green show the matching by normal deviation, Method w.

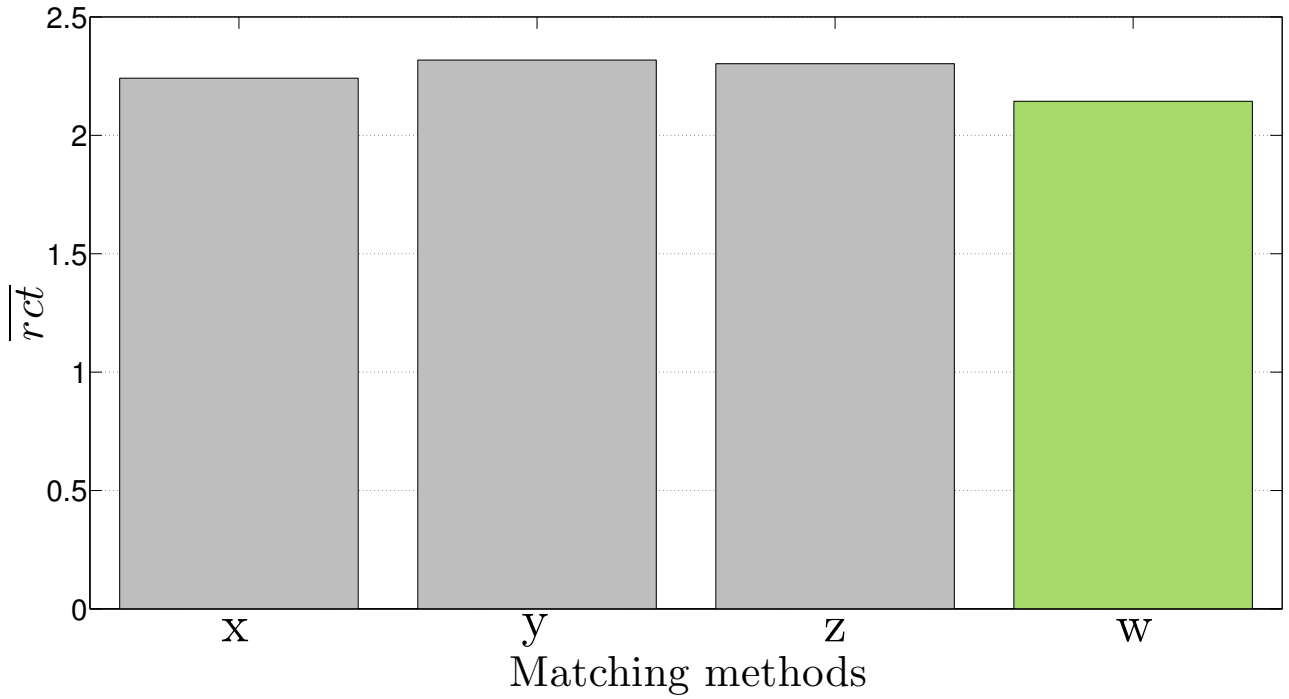


Figure 5.17: Mean relative computational time, $\overline{rct}$, for point matching methods. Bars coloured in green show the matching by normal deviation, Method w.

5.8.3 Unilateral eigentropy rejection

Unilateral eigentropy rejection (Method ϵ) is used for point rejection. Figure 5.10 indicates that among the accurate variants, the combination of angular deviation rejection and distance rejection adjusted (Method $\gamma\delta$) is the most prevalent rejection strategy appearing in 24 % of the accurate variants.

This result is consistent with the conclusions of Chapter 4, however, the method with the best average accuracy is that which combines angular deviation rejection, unilateral eigentropy rejection and distance rejection adjusted by variance (Method $\gamma\delta\epsilon$). Figure 5.18 and 5.19 show that angular deviation rejection (Method γ) and unilateral eigentropy rejection (Method ϵ) are the only methods that appear in the accurate variant set when implemented alone and these two methods appear in all of the combined rejection methods. From the perspective of accuracy, this make a compelling argument for the use of these two methods together for point rejection.

Figure 5.11 shows that angular deviation rejection (Method γ) obtains the best precision performance in occurrence and average precision of the most precise variants. Unilateral eigentropy rejection (Method ϵ) doesn't appear to contribute to the precision of the ICP variants. Figure 5.12 shows that unilateral eigentropy rejection (Method ϵ) appears more often in the efficient ICP variants.

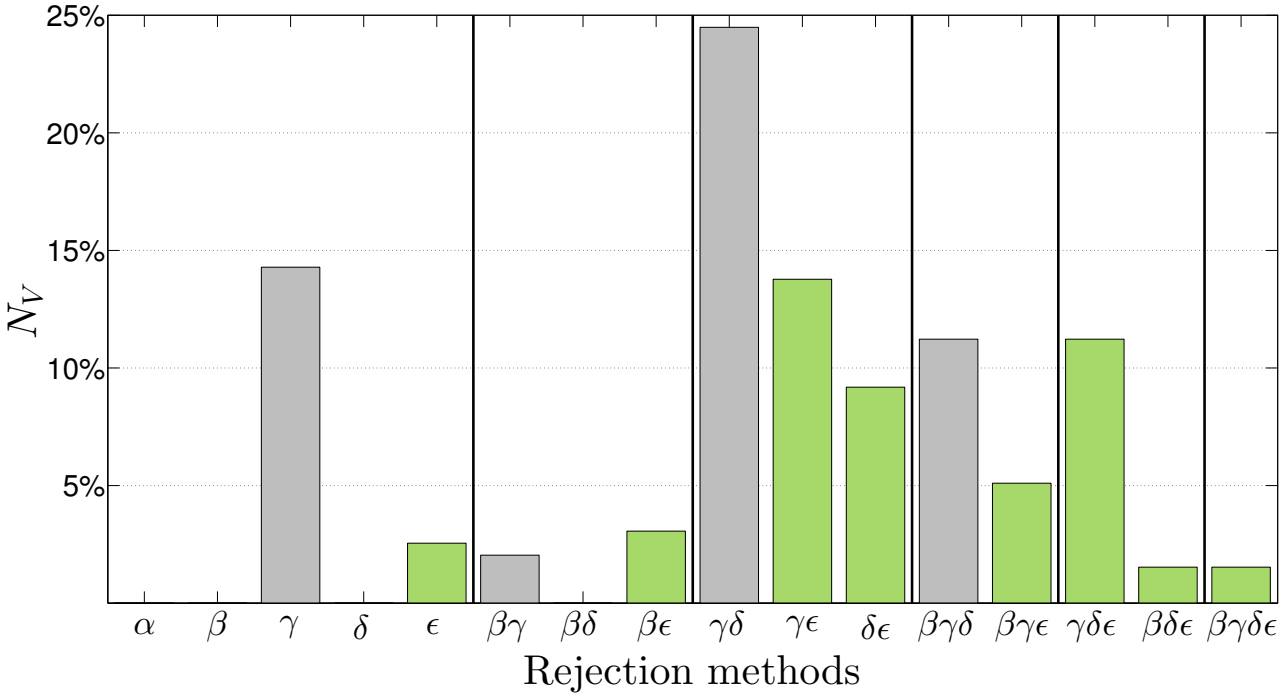


Figure 5.18: Occurrence, N_V, of pair rejection methods within the accuracy tolerance. Bars coloured in green show methods with unilateral eigentropy.

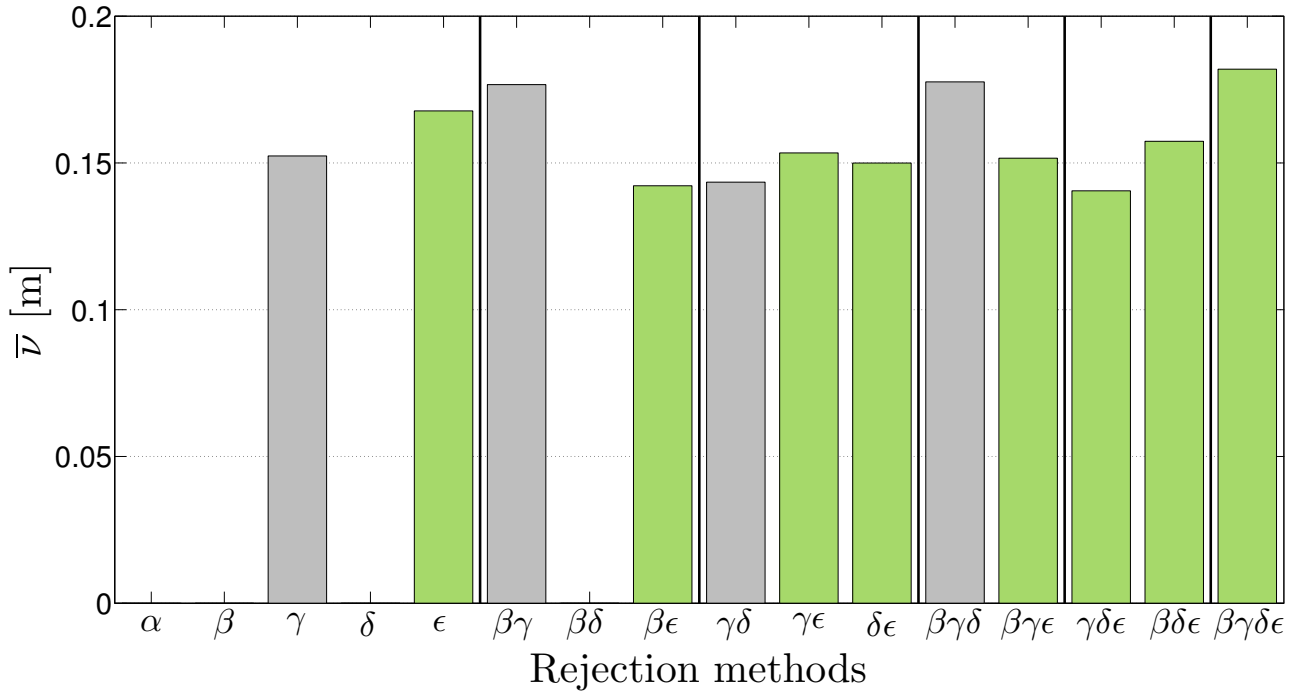


Figure 5.19: Mean accuracy, $\overline{\nu}$, for pair rejection methods. Bars coloured in green show methods with unilateral eigentropy.

5.8.4 Variants that are accurate, precise, and efficient

A set of five ICP variants exist that are accurate, precise, and efficient. Every variant shown in Fig. 5.20, uses all three methods introduced in this chapter. The most accurate variant is composed of: outlier removal (Method b) and eigentropy filter (Method g) for the point selection stage, constant neighbourhood (Method 1) for the neighbourhood selection stage, matching by normal deviation (Method w) for the point matching stage, distance weighting (Method 1) for the pair weighting stage, angular deviation rejection (Method c) with unilateral eigentropy rejection (Method ϵ) for the rejection stage and point-to-plane minimization (Method 1) for the minimization stage.

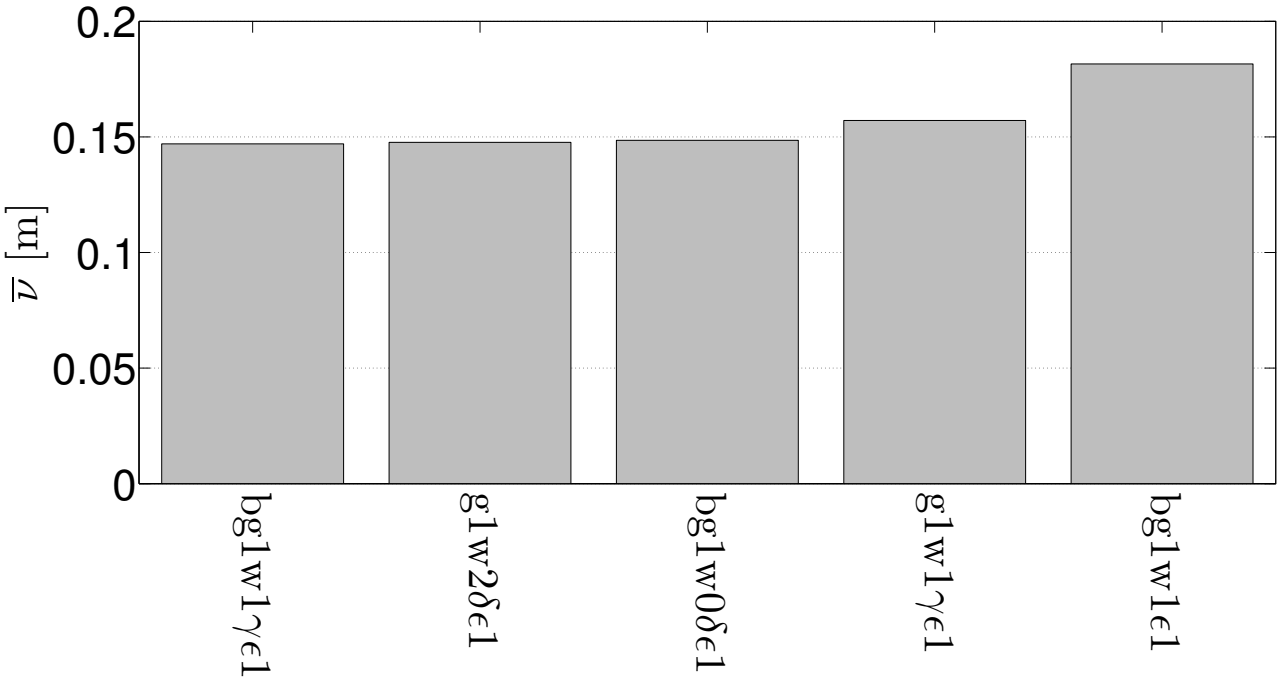


Figure 5.20: Methods in the interception of the most accurate, precise and fast ICP variants. The variants are ordered by average accuracy.

The performance of these methods for accuracy, precision, and efficiency for the three scenes are shown in Fig. 5.21 and 5.22. Significantly these five methods all meet the criteria in each category. The significance of this point relates to the fundamental limitation of many ICP variants in that they work well for some contexts, yet perform poorly in others. There is a need for methods that perform well across the range of contexts to which they are exposed.

The fact that the five variants are accurate, precise, and computational efficient on all data sets represents an advance towards the goal of context independent scan matching by ICP. The results are not fully comprehensive, and are open to challenge that they merely fit the data they have been exposed to. However, they are more general than any other study that has, to the author knowledge, been conducted.

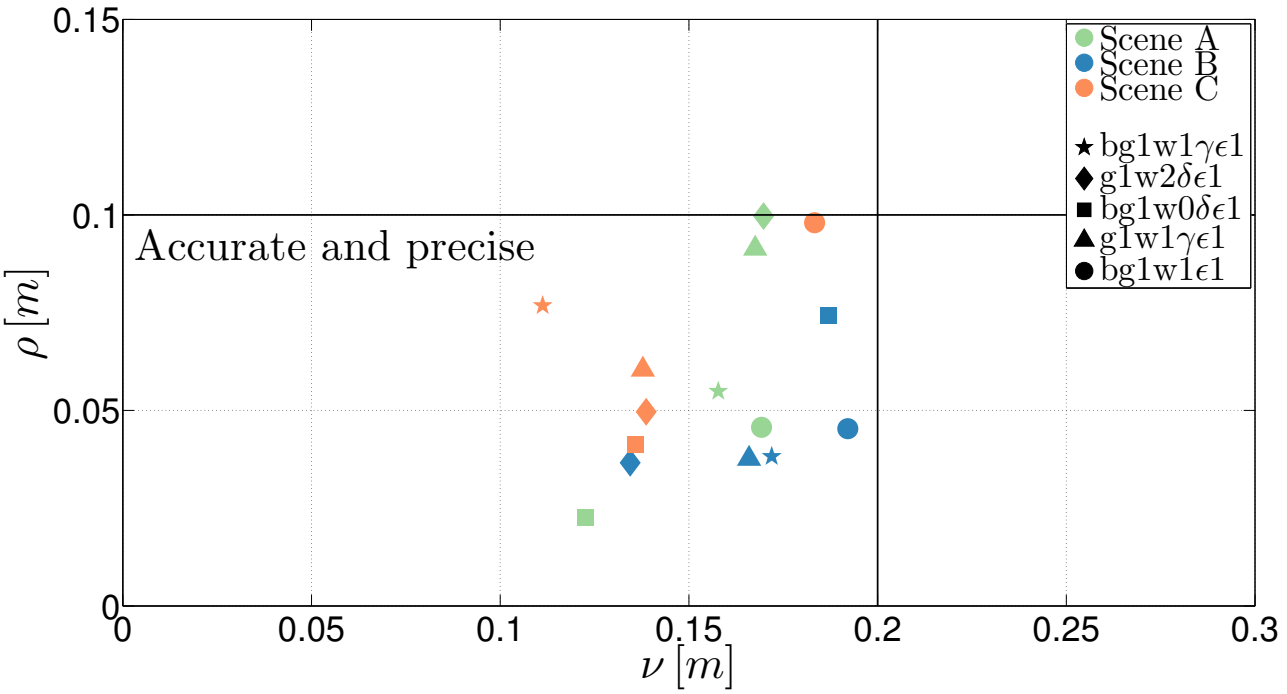


Figure 5.21: Plot of precision, ρ, verses accuracy, ν, of the most accurate, precise and efficient variants. Variants coloured by scene.

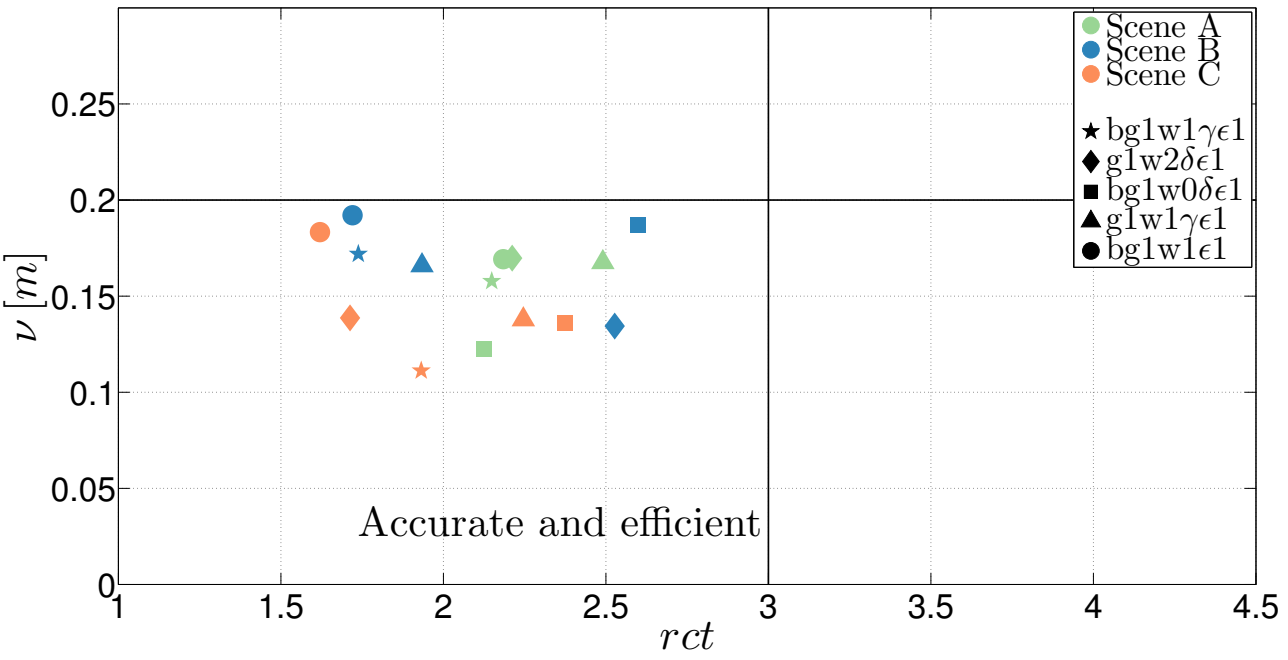


Figure 5.22: Plot of accuracy, ν, verses computational efficiency, rct, of the most accurate, precise and efficient variants. Variants coloured by scene.

5.9 Conclusions

Three methods that act at different parts of the ICP computation were introduced in this chapter. The matching by normal deviation method showed improve both accuracy and precision of ICP variants in the selected scenes. The approach of pairing normals is analogous to matching similar surfaces rather than nearest points. However, this method is not expected to work in all environments, e.g. it may lead to divergence in indoor scenes with long symmetric corridors where many points will have similar normals making unambiguous matching difficult. The methods work here precisely because the normals at points of the clouds are widely distributed. This method uses a fixed number of nearest neighbour deviation, however, the number of neighbours could be set dynamically to further improve performance using, for example, the eigentropy of the point.

Establishing a performance trend for point selection methods is not clear. However, there are some combinations that produce a poor performance, for example when density filter and dimensionality selection are used. In the same way there are others that produce good results, as with the combination of outlier removal with geometrical stable sampling. The eigentropy filter exhibits a good overall accuracy alone or in combination with any other point selection method for the data sets considered.

Importantly, the eigentropy filter is faster than stable covariance sampling, and this is crucial to providing accurate, precise and efficient ICP variants when the eigentropy filter method is involved. The most accurate, precise and efficient variants all use the eigentropy filter alone or in combination with outlier removal.

Pair rejection methods clearly improve the performance of the ICP algorithm. ICP variants with no pair rejection strategy do not appear in the most accurate variants. Angular deviation rejection outperforms other rejection methods and in combination with distance rejection adjusted by variance, gives the best performance overall.

Unilateral eigentropy rejection and distance rejection adjusted by variance work as complements of other stronger rejection strategies.

The most accurate, precise and efficient ICP variants include the unilateral eigentropy rejection. This result indicates that unilateral eigentropy rejection can improve the overall performance of ICP variants but it is less significant than matching by normal deviation and the application of the eigentropy filter for point selection.

The overall conclusion of this chapter is that the three methods introduced in this chapter improve the performance of scan matching for the data sets considered.

Chapter 6

Conclusions and future work

The work conducted in this thesis addressed the issue of terrain mapping in open-cut mining environments using sensors on moving platforms. The focus of the thesis has been on variants of the iterative closest point algorithm with the aim of improving frame-by-frame scan matching. This is a form of LiDAR odometry. This chapter summaries the thesis, drawing conclusions on the findings, and suggesting directions for future work.

6.1 Thesis summary

In essence, the work of this thesis can be summarised as:

1. Experimental work to obtaining sensor data in a real mining environment appropriate to the thesis problem. This work delivered three representative scenes of open-cut mining environments.
2. A systematic review and comparison of publish iterative closest point variants.
3. A comparative investigation of three novel methods based on the geometrical information content of point clouds for improving the performance of the ICP algorithm.

Chapter 4 compared different published variants of the ICP algorithm against a data set typical of that for open-pit mining. This evaluation reveals that the performance of ICP variants is strongly influenced by point distributions and the geometrical structure of the point cloud. No single variant could be found that simultaneously fulfils the requirements presented for accuracy, precision and efficiency against the three scene datasets used for this work. Specific conclusions for each stage of the ICP computation are as follows.

Minimization

Minimization by point-to-plane was found to outperform other minimization methods. Point-to-point minimization was found to perform poorly for the terrain mapping problem. Generalized minimization is not sensitive to the methods used for different stages of the ICP algorithm.

Matching

The nearest neighbour algorithm enhanced by normal was found to beneficially improve the accuracy of ICP variants.

Rejection

Rejection methods improve the performance of ICP variants. Rejection by angular deviation gives the best overall performance.

Neighbourhood selection

Dynamic neighbourhood selection beneficially influence the accuracy performance of ICP variants. The bounded radius method was found to be the best neighbourhood selection method, however, it is computationally expensive making it unattractive for real-time applications.

Point selection

No clear preference for point selection for the many alternative point selection methods was found.

Weighting

The influence of weighting is not significant along variants, however the choice of a correct weighting may produce a slight improvement on registration performance.

Chapter 5 introduced the "eigentropy" feature for quantifying the geometrical disorder of a point cloud. Furthermore, three novel methods for complementing the ICP algorithm have been presented. These methods are termed: eigentropy filter, matching by normal deviation and unilateral eigentropy rejection. These novel methods were added to the set of methods presented in Chapter 4 and evaluated using the same performance methodology. This evaluation concludes that these novel methods can improve the overall performance of the ICP algorithm. This improvement resulted in the identification of five ICP variants that simultaneously fulfilled the prescribed requirements for accuracy, precision and computational efficiency. Specific conclusions for each new method are:

Eigentropy filter

This method improves the accuracy of ICP variants and is faster than methods with similar accuracy performance. The eigentropy filter is present in all of the ICP variants that simultaneously meet the performance requirements in accuracy, precision and computational efficiency.

Matching by normal deviation

This method outperforms other matching methods in accuracy, precision and computational efficiency. Matching by normal deviation is present in all of the ICP variants that simultaneously meet performance requirements in accuracy, precision and computational efficiency.

Unilateral eigentropy rejection

This method provides very good performance when it is used with other rejection methods. Some variants may present worse performance than what would be obtained without any rejection strategy. However, unilateral eigentropy rejection is present in all of the ICP variants that simultaneously meet performance requirements in accuracy, precision and computational efficiency.

6.2 Thesis contributions

The original contributions of this thesis are:

1. The establishment of a methodology that evaluates the performance of registration methods in the context of terrain mapping. Moreover, it has presented a novel precision method to quantify the "deviation" error of consecutive scan matching.

2. The comparison of thousands of ICP variants by finding valuable statistical information in the performance of different methods of the ICP computation process. Furthermore, it has presented the first study of performance of *dynamic neighbourhood selection* in the context of scan registration.

3. The introduction of "eigentropy" features for quantifying the geometrical information of a point in a similar spirit to the entropy measure from thermodynamics and information theory.

4. The development of three novel methods which considerably improve the performance of the ICP algorithm.

6.3 Potential caveats

The principal limitation of this study is that dynamic environments are not included in the data set. A simple approach to deal with dynamic environments is to understand the moving objects in the scene as areas of non-overlapping between consecutive scans. This means that these points could be "meaningfully ignored" by different rejection methods. Another approach is detecting and tracking the moving object in order to filter the points associated with them. This approach requires specialized methods that are out of the scope of this thesis.

Notwithstanding, the performance of the ICP variants evaluated in this thesis may be similar in dynamic environments if is possible rejecting or filtering the moving objects.

6.4 Recommendations for future work

The biggest challenge for registration methods is the need to be robust against different geometries present in the environment. This thesis has shown that the distribution of points in a point cloud is the principal factor for the dissimilar performance of ICP variants. The key issue is to obtain quantifiable information about the distribution of points in order to adapt the registration method to the geometry of the scene. This problem has two dimensions: i) the adjustment of internal parameters of different methods; and ii) the selection of methods based on their performance in different scenes.

The majority of methods have internal parameters that need to be set prior by a user. However, there are methods that dynamically adjust internal parameters. An example of a method that adjusts its parameters by the distribution of point is the "distance rejection adjusted by the variance of the distance between matched points"[1]. This method has shown a notable improvement in the performance of ICP variants when it is applied as a complement to other rejection methods. The same idea may be expanded to other methods, for example, analysing the eigentropy of points in order to adapt parameters by a direct quantification of the distribution of points.

Selecting methods depending on the geometry of the environment follows the same idea. There are some methods than can be used, for example, in structured indoor environments and others in natural landscapes. These strengths will definitely aid in reducing the gap to reach fully autonomous mapping.

6.5 Concluding remarks

Scan matching is only part of the process of mapping terrain. Having transformed individual scan point clouds into a common frame of reference, further reduction is needed for the information to be useful to an automation system. The depth and complexity of this topic is not, at all, touched on in this work. Scan matching is however a *sine qua non* for effective mapping without the use of an independent navigation system and hence it can be considered a foundation capability. The work of this thesis has been motivated first and foremost by the need for improvement in the algorithms for scan matching for environments such as open-cut mining where the terrain is loosely structured.

A criticism that can be levelled at previous works using ICP for scan matching is that the methods that have emerged over the last 20 years have been crafted for the data that they are evaluated against. For this reason, the methods often perform poorly when applied to data that is different. This is fair

[1] Method δ in Chapter 4.

criticism and it is a criticism that can be directed at the work of this thesis also. While the major finding of the thesis was the identification of five variants that are accurate, precise and computational efficient over the three scenes typical of open-cut mining, the evaluation has only been over three scenes, each comprising 100 scans.

The defence against this criticism is one of degree. The work has been significantly more comprehensive in the range of data it has considered and the level of rigour applied in that analysis. The findings, while limited in their generality, present an advance in the state of the art of scan matching by ICP in so far as the results are more accurate, more precise, more efficient, and more robust than what has come before. The baton is accordingly now passed to others to further improve on the methods for solving this problem.

Bibliography

Devrim Akca. Registration of point clouds using range and intensity information. In *The International Workshop on Recording, Modeling and Visualization of Cultural Heritage*, pages 115–126, 2005.

Applanix Corp. POS LV 420 V4. Data sheet, Applanix Corp., 2008.

K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (5):698–700, 1987.

Hernan Badino, Daniel Huber, Yongwoon Park, and Takeo Kanade. Fast and accurate computation of surface normals from range images. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3084–3091. IEEE, 2011.

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.

Jens Behley, Volker Steinhage, and Armin B Cremers. Performance of histogram descriptors for the classification of 3d laser range data in urban environments. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4391–4398. IEEE, 2012.

Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

Peter Biber and Wolfgang Straßer. nscan-matching: Simultaneous matching of multiple scans and application to slam. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2270–2276. IEEE, 2006.

Nicola Brusco, Marco Andreetto, Andrea Giorgi, and Guido Maria Cortelazzo. 3d registration by textured spin-images. In *3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on*, pages 262–269. IEEE, 2005.

Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.

Ingemar J Cox. Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. *Robotics and Automation, IEEE Transactions on*, 7(2):193–204, 1991.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

Jérôme Demantké, Clément Mallet, Nicolas David, and Bruno Vallet. Dimensionality based scale selection in 3d lidar point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Laser Scanning*, 2011.

Tamal K Dey, Gang Li, and Jian Sun. Normal estimation for point clouds: A comparison study for a voronoi based method. In *Point-based graphics, 2005. Eurographics/IEEE VGTC symposium proceedings*, pages 39–46. IEEE, 2005.

Yago Diez, Joan Martí, and Joaquim Salvi. Hierarchical normal space sampling to speed up point cloud coarse matching. *Pattern Recognition Letters*, 33(16):2127–2133, 2012.

Albert Diosi and Lindsay Kleeman. Laser scan matching in polar coordinates with application to slam. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3317–3322. IEEE, 2005.

E. Duff. Tracking a vehicle from a rotation platform with a scanning range laser. In *Proc. Australasian Conference on Robotics and Automation*, December 2006.

Jan Elseberg, Stéphane Magnenat, Roland Siegwart, and Andreas Nüchter. Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration. *Journal of Software Engineering for Robotics*, 3(1):2–12, 2012.

Raúl San José Estépar, Anders Brun, and Carl-Fredrik Westin. Robust generalized total least squares iterative closest point registration. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2004*, pages 234–241. Springer, 2004.

Maurice F Fallon, Hordur Johannsson, Jonathan Brookshire, Seth Teller, and John J Leonard. Sensor fusion for flexible human-portable building-scale mapping. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4405–4412. IEEE, 2012.

FARO Technologies Inc. *FARO Laser Scanner Focus^{3D}*, 2010.

Jacques Feldmar, Nicholas Ayache, and Fabienne Betting. 3d-2d projective registration of free-form curves and surfaces. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 549–556. IEEE, 1995.

Torsten Fiolka, Jörg Stückler, Dominik A Klein, Dirk Schulz, and Sven Behnke. Sure: Surface entropy for distinctive 3d features. In *Spatial Cognition VIII*, pages 74–93. Springer, 2012.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

Alex Flint, Anthony R Dick, and Anton Van Den Hengel. Thrift: Local 3d structure recognition. In *dicta*, volume 7, pages 182–188, 2007.

Tom Ford, Jason Hamilton, Mike Bobye, and Laurence Day. Gps/mems inertial integration methodology and results. In *Proceedings of the 17th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2004)*, pages 1587–1597, 2001.

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

Natasha Gelfand, Leslie Ikemoto, Szymon Rusinkiewicz, and Marc Levoy. Geometrically stable sampling for the icp algorithm. In *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*, pages 260–267. IEEE, 2003.

Guy Godin, Marc Rioux, and Rejean Baribeau. Three-dimensional registration using range and intensity information. In *Photonics for Industrial Applications*, pages 279–290. International Society for Optics and Photonics, 1994.

Adrien Gressin, Clément Mallet, and Nicolas David. Improving 3d lidar point cloud registration using optimal neighborhood knowledge. *Proceedings of ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Melbourne, Australia*, 5:111–116, 2012.

Adrien Gressin, Clément Mallet, Jérôme Demantké, and Nicolas David. Towards 3d lidar point cloud registration improvement using optimal neighborhood knowledge. *ISPRS Journal of Photogrammetry and Remote Sensing*, 79:240–251, 2013.

Marcus Hebel and Uwe Stilla. Automatic registration of laser point clouds of urban areas. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36 (3/W49A):13–18, 2007.

R. Hoffman and A.K. Jain. Segmentation and classification of range images. *IEEE Trans Pattern Anal Mach Intell.*, 9(6):606–620, 1987.

H Houshiar, Jan Elseberg, Dorit Borrmann, and Andreas Nüchter. A study of projections for key point based registration of panoramic terrestrial 3d laser scans. *Geo-spatial Information Science (GSIS)*, 2013.

J. Huang and C.H. Menq. Automatic data segmentation for geometric feature extraction from unorganized 3-d coordinate points. *IEEE Trans Robotics Autom,*, 17(3):268–279, 2001.

Benjamin Huhle, Martin Magnusson, Wolfgang Straßer, and Achim J Lilienthal. Registration of colored 3d point clouds with a kernel-based extension to the normal distributions transform. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 4025–4030. IEEE, 2008.

Anil K Jain and Chitra Dorai. 3d object recognition: Representation and matching. *Statistics and Computing*, 10(2):167–182, 2000.

Andrew Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.

Lyle Johnson and Frank Van Diggelen. Advantages of a combined gps+ glonass precision sensor for machine control applications in open pit mining. In *Position Location and Navigation Symposium, IEEE 1998*, pages 549–554. IEEE, 1998.

Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

Timor Kadir and Michael Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

K Khoshelhama, DR Dos Santos, and G Vosselman. Generation and weighting of 3d point correspondences for improved registration of rgb-d data. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(2):127–132, 2013.

Alireza Khotanzad and Yaw Hua Hong. Invariant image recognition by zernike moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(5):489–497, 1990.

Klaas Klasing, Daniel Althoff, Dirk Wollherr, and Martin Buss. Comparison of surface normal estimation methods for range sensing applications. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3206–3211. IEEE, 2009.

G. Kloos, J. Guivant, S. Worrall, A. Maclean, and E. Nebot. Wireless network for mining application. In *Australasian Conference on Robotics and Automation (ACRA) 2004*, December 2004.

J-F Lalonde, Ranjith Unnikrishnan, Nicolas Vandapel, and Martial Hebert. Scale selection for classification of point-sampled 3d surfaces. In *3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on*, pages 285–292. IEEE, 2005.

Jean-François Lalonde, Nicolas Vandapel, Daniel F Huber, and Martial Hebert. Natural terrain classification using three-dimensional ladar data for ground robot mobility. *Journal of field robotics*, 23(10):839–861, 2006.

David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

Feng Lu and Evangelos Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 18(3):249–275, 1997.

Martin Magnusson and Tom Duckett. A comparison of 3d registration algorithms for autonomous underground mining vehicles. In *Proceedings of the European Conference on Mobile Robotics (ECMR 2005)*, pages 86–91, 2005.

Takeshi Masuda and Naokazu Yokoya. A robust method for registration and segmentation of multiple range images. *Computer Vision and Image Understanding*, 61(3):295–307, 1995.

Gerard Medioni, Mi-Suen Lee, and Chi-Keung Tang. *A computational framework for segmentation and grouping*. Elsevier, 2000.

Hao Men, Biruk Gebre, and Kishore Pochiraju. Color point cloud registration with 4d icp algorithm. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1511–1516. IEEE, 2011.

Niloy J Mitra, An Nguyen, and Leonidas Guibas. Estimating surface normals in noisy point cloud data. *International Journal of Computational Geometry & Applications*, 14(04n05):261–276, 2004.

DJ Natale, MS Baran, and Richard L Tutwiler. Point cloud processing strategies for noise filtering, structural segmentation, and meshing of ground-based 3d flash lidar images. In *Applied Imagery Pattern Recognition Workshop (AIPR), 2010 IEEE 39th*, pages 1–8. IEEE, 2010.

Andreas Nüchter. *3D robotic mapping: the simultaneous localization and mapping problem with six degrees of freedom*, volume 52. Springer, 2009.

Andreas Nuechter, Kai Lingemann, and Dorit Borrmann. The 3d toolkit, 2012. URL `http://sourceforge.net/projects/slam6d/files/`.

Zhen Peng. Efficient matching of robust features for embedded slam. Master's thesis, Universitat Stuttgart, Holzgartenstr. 16, 70174 Stuttgart, 2012. URL `http://elib.uni-stuttgart.de/opus/volltexte/2012/7617`.

TG Phillips, ME Green, and PR McAree. An adaptive structure filter for sensor registration from unstructured terrain. *Journal of Field Robotics*, 2014.

Birgit M Planitz, Anthony J Maeder, and JA Williams. The correspondence framework for 3d surface matching algorithms. *Computer Vision and Image Understanding*, 97(3):347–383, 2005.

François Pomerleau, Francis Colas, Roland Siegwart, and Stéphane Magnenat. Comparing icp variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013.

Kari Pulli. Multiview registration for large data sets. In *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*, pages 160–168. IEEE, 1999.

Zheng Qin, Ji Jia, and Jun Qin. Content based 3d model retrieval: A survey. In *Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on*, pages 249–256. IEEE, 2008.

Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5, 2009.

Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24(4):345–348, 2010.

Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz. Persistent point feature histograms for 3d point clouds. In *Proc 10th Int Conf Intel Autonomous Syst (IAS-10), Baden-Baden, Germany*, pages 119–128, 2008.

Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.

Juan Manuel Sáez, Andrew Hogue, Francisco Escolano, and Michael Jenkin. Underwater 3d slam through entropy minimization. In *Robotics and automation, 2006. ICRA 2006. Proceedings 2006 IEEE international conference on*, pages 3562–3567. IEEE, 2006.

Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5):578–596, 2007.

Oliver Schall, Alexander Belyaev, and Hans-Peter Seidel. Robust filtering of noisy scattered point data. In *Proceedings of the Second Eurographics/IEEE VGTC conference on Point-Based Graphics*, pages 71–77. Eurographics Association, 2005.

C Schutz, T Jost, and H Hugli. Multi-feature matching algorithm for free-form 3d surface registration. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 2, pages 982–984. IEEE, 1998.

Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia*, pages 357–360. ACM, 2007.

Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: Science and Systems*, volume 2, page 4, 2009.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948.

Gregory C Sharp, Sang W Lee, and David K Wehe. Icp registration using invariant features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(1):90–102, 2002.

Øystein Skotheim, Morten Lind, Pål Ystgaard, Sigurd Fjerdingen, et al. A flexible 3d object localization system for industrial part handling. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3326–3333. IEEE, 2012.

Federico M Sukno, John L Waddington, and Paul F Whelan. Comparing 3d descriptors for local search of craniofacial landmarks. In *Advances in Visual Computing*, pages 92–103. Springer, 2012.

Johan WH Tangelder and Remco C Veltkamp. A survey of content based 3d shape retrieval methods. *Multimedia tools and applications*, 39(3):441–471, 2008.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

Andrea Torsello, Emanuele Rodola, and Andrea Albarelli. Sampling relevant points for surface registration. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, pages 290–295. IEEE, 2011.

Yanghai Tsin and Takeo Kanade. A correlation-based approach to robust point set registration. In *Computer Vision-ECCV 2004*, pages 558–569. Springer, 2004.

Ranjith Unnikrishnan, J-F Lalonde, Nicolas Vandapel, and Martial Hebert. Scale selection for the analysis of point-sampled curves. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 1026–1033. IEEE, 2006.

Velodyne LiDAR Inc. *HDL-64E S2 and S2.1: High definition LiDAR sensor*. 345 Digital Drive, Morgan Hill, CA 95037, 2008.

Gerhard Weiß and Ewald von Puttkamer. A map based on laserscans without geometric interpretation. In *Intelligent Autonomous Systems*, volume 4, pages 403–407, 1995.

Karen F West, Brian N Webb, James R Lersch, Steven Pothier, Joseph M Triscari, and A Evan Iverson. Context-driven automated target detection in 3d data. In *Defense and Security*, pages 133–143. International Society for Optics and Photonics, 2004.

Thomas Wiemann, Andres Nüchter, Kai Lingemann, Stefan Stiene, and Joachim Hertzberg. Automatic construction of polygonal maps from point cloud data. In *Proc. 8th IEEE Intl. Workshop on Safety, Security, and Rescue Robotics (SSRR-2010), Bremen, Germany*, page 22, 2010.

Jianjie Wu and Qifu Wang. Feature point detection from point cloud based on repeatability rate and local entropy. In *International Symposium on Multispectral Image Processing and Pattern Recognition*, pages 67865H–67865H. International Society for Optics and Photonics, 2007.

Hui Xie, Kevin T McDonnell, and Hong Qin. Surface reconstruction of noisy and defective data sets. In *Visualization, 2004. IEEE*, pages 259–266. IEEE, 2004.

M. Yang and E. Lee. Segmentation of measured point data using a parametric quadric surface approximation. *Comput-Aided Des.*, 31(6):449–457, 1999.

Jian Yao, Mauro R Ruggeri, Pierluigi Taddei, and Vítor Sequeira. Automatic scan registration using 3d linear and planar features. *3D Research*, 1(3):1–18, 2010.

Peng Ye and Fang Liu. Novel image registration method using multiple gaussian mixture models. In *Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on*, pages 2117–2120. IEEE, 2012.

Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13(2):119–152, 1994.

Zhengyou Zhang and Olivier D Faugeras. Determining motion from 3d line segment matches: a comparative study. *Image and Vision Computing*, 9(1):10–19, 1991.